

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tanja Miklič

Metodologija migracije podatkov

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVA IN
INFORMATIKE

Ljubljana, 2016

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Tanja Miklič

Metodologija migracije podatkov

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVA IN
INFORMATIKE

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Metodologija migracije podatkov

Tematika naloge:

Migracija podatkov je potrebna pri večini projektov razvoja informacijskih sistemov ali projektih implementacije ERP in/ali CRM sistemov. V praksi se migracija lahko izkaže kot zelo kompleksen podprojekt v okviru projekta. V diplomski nalogi predstavite področje migracije in vidike migracije. Na podlagi tega in vaših izkušenj v praksi opredelite metodologijo migracije podatkov. V okviru opredelitve metodologije opredelite tudi vloge, ki nastopajo v okviru procesa migracije.

IZJAVA O AVTORSTVU ZAKLJUČNEGA DELA

Spodaj podpisana Tanja Miklič, vpisna številka 63990276, avtor zaključnega dela z naslovom:

Metodologija migracije podatkov (angl. *The methodology of data migration*)

IZJAVLJAM

1. da sem pisno zaključno delo študija izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil/-a;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

V Ljubljani, dne 18. avgusta 2016

Podpis študenta/-ke:

Zahvaljujem se doc. dr. Roku Rupniku za pomoč in uporabne nasvete.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija za temo	2
2	Migracija podatkov	3
2.1	Definicija	3
2.2	Značilnosti migracije podatkov	4
3	Metode migracije podatkov	9
3.1	Tehnični vidik	10
3.2	Vodstveni vidik	14
4	Metodologija migracije podatkov	43
4.1	Faza 1: Inicializacija	44
4.2	Faza 2: Analiza	61
4.3	Faza 3: Razvoj	72
4.4	Faza 4: Testiranje	76
4.5	Faza 5: Izvedba	82
5	Sklepne ugotovitve	89
	Literatura	91

Seznam uporabljenih kratic

kratica	angleško	slovensko
ETL	extract, transform and load	izvoz, preoblikovanje, vnos
IT	information technology	informacijska tehnologija
SQL	structured query language	strukturiran povpraševalni jezik za delo s podatkovnimi bazami
API	application program interface	programski vmesnik

Povzetek

Naslov: Metodologija migracije podatkov

Namen diplomske naloge je predstavitev različnih metod in strategij pri migraciji podatkov iz nekega vira podatkov v (praviloma) podatkovno bazo, opisati vloge strokovnjakov na takem projektu, opozoriti na možne zaplete in ponuditi rešitve, ki jih je mogoče najti v literaturi ali so del pridobljenih izkušenj.

Ker se migracija podatkov razlikuje od projekta do projekta, je obravnavana tema predstavljena splošno, da bi tako zaobjeli čim večji obseg projektov in morda s tem olajšali pot nekomu, ki se bo srečal z zahtevo po migraciji podatkov.

Tema je vedno vnovičen izziv, ker se poslovne aplikacije nenehno spreminjajo ali celo menjujejo (nove tehnologije, nova poslovna pravila), količina generiranih podatkov se povečuje, lastniki poslovnih podatkov pa s preходом na nov informacijski sistem ne želijo izgubiti starih podatkov.

Ključne besede: migracija podatkov, podatki, metodologija, metodologija migracije podatkov.

Abstract

Title: The methodology of data migration

The aim of the thesis is to present a list of different data migration methods and strategies from some data source to (usually) database, to describe roles experts have on such a project, to warn of the potential complications and to offer solutions, which can be found in the literature or are part of experience gained on different projects.

The topic is presented in general, since nature of data migration varies from project to project, in order to embrace as many projects as possible and to help anyone who will be faced with data migration.

The theme is particularly interesting because the business applications are constantly changing or are even replaced with new ones (new technologies, new business rules), the amount of generated data is increasing, the owners of business data do not want to lose the old data while data is transitioned to a new information system.

Keywords: Data Migration, Data, Methodology, The methodology of data migration.

Poglavje 1

Uvod

Z informatizacijo poslovnih sistemov so podjetja postala konkurenčnejša. Vendar, da bi obdržala to konkurenčnost, se srečujejo z izzivom prenove sistema v današnjem okolju, ki se zelo hitro poslovno in tehnološko spreminja. Razlogi za zamenjavo ali nadgradnjo sistema so:

- Zaradi nefleksibilnosti trenutnega sistema se nove poslovne funkcionalnosti, ki bi ohranile konkurenčnost, ne morejo enostavno dodati.
- Računalniška strojna oprema, na kateri deluje sistem, je zastarela, draga za vzdrževanje ter počasna.
- Zaradi pomanjkljive dokumentacije in slabega poznavanja sistema je vzdrževanje sistema drago, iskanje napak pa je časovno potratno.
- Zaradi nepopolnega in zapletenega vmesnika je vložek za integracijo z drugimi sistemi visok.

Dolžina časovnega obdobja, v katerem se sistem lahko zamenja ali nadgradi, je odvisna od kompleksnosti sistema. Vsekakor sistema ni mogoče prenoviti v enem dnevu. Razumevanje, kaj sistem počne, je lahko dolgotrajno, tudi če nas ne zanima, kako to počne. Tako je, ko so zahteve natančno določene, razvoj novega sistema verjetno najenostavnejši del prenove. Z dolgoletno

uporabo sistema so se nakopičili tudi podatki, ki jih podjetje ponavadi s prehodom na nov sistem ne želi izgubiti. Ker je danes zaželeno, da je preklop na nov sistem hiter in je tako tudi vpliv na poslovno okolje minimalen, mora biti strategija za preklop sistemov dobro premišljena. Na koncu ne smemo pozabiti na obsežno testiranje ter seveda naročnikovo validacijo in njegov prevzem sistema.

1.1 Motivacija za temo

Vsaka migracija podatkov je zahteven, od sistema do sistema unikaten proces, zato je potrebno pričujoč dokument vzeti bolj kot niz opornih točk pri posameznih korakih migracije, saj je od projekta odvisno, kakšni so obseg migracije, občutljivost podatkov, časovna in stroškovna komponenta, uporabljena orodja in nenazadnje tudi naročnik. Če je naročnik dobro tehnično izobražen, je to po eni strani prednost, saj imamo dobrega sogovornika, po drugi strani pa lahko od takega naročnika dobimo zahteve, ki morda niso v skladu z našo prakso dela, morda moramo vložiti dodaten napor za raziskavo, izvedbo in testiranje zahtevane rešitve. Če naročnik ni tehnično izobražen, imamo sicer večjo svobodo pri tehničnih odločitvah, vendar pa od takega naročnika ne moremo pričakovati konstruktivnih pripomb.

V začetku dela so predstavljene definicije in značilnosti migracije podatkov. Sledi seznam metod in strategij. Na koncu je še podrobno opisan potek migracije podatkov v praksi z vsemi izvedenimi koraki. Čeprav je vsaka migracija edinstvena, pa njena izvedba, vsaj do neke mere, sledi določenemu zaporedju korakov. Delo je bilo napisano z namenom, da se vsaj delno olajša pot nekomu, ki bo zadolžen za migracijo podatkov in se bo srečal vsaj z nekaterimi problemi, opisanimi v poglavju Metodologija migracije podatkov.

Poglavje 2

Migracija podatkov

2.1 Definicija

“Migracija podatkov je potrebna, ko se zamenja informacijski sistem oziroma se ga nadgradi ali ko se dva ali več sistemov združi v enega (eno podjetje prevzame drugo in želi imeti usklajene, združene podatke).” [17]

Po mnenju Iduja [7] obstaja malo akedemskih objav, ki bi podale definicijo izraza. Večina literature se ostredotoča le na določen del migracije, ne pa na proces kot celoto. Tako je Drumm podal prvo definicijo izraza leta 2007: “Migracija podatkov je definirana kot proces preoblikovanja in združevanja podatkov z ene ali več podedovanih podatkovnih skladišč na novo ciljno podatkovno skladišče. Med procesom migracije se morajo podatki pridobiti iz vira, se transformirati in naložiti na cilj.”

“Migracija podatkov je enkraten, aplikacijsko podprt proces, ki prenese podatke z izvora, ki naj bi se predvidoma umaknil iz sistema, na ciljno aplikacijo, ki ima ponavadi različen podatkovni model.” [2]

Ker je migracija podatkov velikokrat enkraten postopek, brez njegovega ponavljanja in pridobljenih izkušenj pa je težko trditi, katera metoda bo omogočila, da bo projekt migracije podatkov uspešno zaključen v časovnem

in stroškovnem okviru, se lahko zanašamo le na znanje tistih, ki imajo veliko izkušenj z migracijo podatkov, ali pa eksperimentiramo sami. Problem je tudi, ker se način migracije podatkov razlikuje od poslovnega sistema do poslovnega sistema.

Kljub vsemu je Matthesu in Schulzu [3] leta 2011 uspelo združiti tako akademsko znanje kot že obstoječe zapise izvajalcev migracije podatkov ter dodati svoje raziskovalno znanje. Njuna definicija migracije podatkov je: “Enkratni process, podprt z orodji, katerega cilj je prenos formatiranih podatkov s strukture na izviru na podatkovno ciljno strukturo, pri čemer se strukturi podatkov razlikujeta na konceptualnem in/ali fizičnem nivoju.” Kot konceptualni nivo so mišljeni poslovni koncepti, ki jih uporablja aplikacija in z njimi modelira njene podatke. Fizični nivo pa predstavlja tehnično rešitev, kje in kako so podatki shranjeni v sistemu.

2.2 Značilnosti migracije podatkov

Če upoštevamo značilnosti migracije podatkov našete spodaj, ki jih je zbral Idu [7], se lahko marsikdaj izognemo nevšečnostim ter zaključimo projekt pravočasno.

1. Migracija podatkov je samostojen IT projekt (Willinger in Gradl, 2004), s svojim začetkom in koncem, ki ga med drugim sestavljajo tri faze:

- Planiranje;
- Implementacija;
- Testiranje (enote, integracijsko);

2. Malo izvedb (Haller, 2008)

Enkratna aktivnost, ki se redko ponavlja, zato tudi znanja in dokumentacije na to temo ni veliko.

3. Visoko tveganje (Informatica, 2010)

Podatki v poslovnem sistemu so veliko vredni, zato je tudi kakršnakoli njihova manipulacija tvegana.

4. Neskončna (Wu in Tang, 1997; Russom, 2006)

V primerjavi z ostalimi projekti se migracija podatkov sicer izvaja malokrat, vendar pa je potrebna zaradi nadgradnje, vpeljave novega sistema ali združevanja dveh ali več poslovnih sistemov, dokler podatki v sistemu obstajajo. Potreba po njej je odvisna od tehnološkega napredka in spreminjanja poslovnega procesa.

5. Tesno povezana s projektom razvoja poslovnega sistema (Morris, 2006)

Migracija podatkov je odvisna od ostalih aktivnosti na projektu razvoja poslovnega sistema in sistem je lahko uspešno dan v uporabo šele, ko prenesemo vse potrebne podatke. Zato morata biti projekt migracije podatkov in projekt poslovnega sistema neprestano usklajena.

6. Tehničen problem, vendar kritična za projekt razvoja poslovnega sistema (Russom, 2006)

Poslovni sistem brez pravilno prenešenih podatkov ni uporaben in čeprav migracija podatkov izgleda kot tehničen problem, je večji izziv razumeti podatke.

7. Več kot le kopiranje podatkov (Russom, 2006)

Podatki se redko samo prekopirajo. V večini primerov se podatki transformirajo, kar lahko vključuje tudi čiščenje podatkov in s tem njihovo večjo kvaliteto.

8. Iterativna (Haller, 2008)

Migracija podatkov je ciklični process. Razumevanje podatkov na izvoru je lahko dolgotrajen proces, zato začnemo s testnimi migracijami na podatkih, ki jih poznamo, in kasneje dodajamo kompleksnost, ko imamo preslikavo tudi za druge podatke, in redno izvajamo planiranje, implementacijo in testiranje.

9. Tema predvsem tistih, ki so migracijo podatkov izvajali v praksi (Haller, 2009; Morris, 2006)

Večina akademskih objav je iz leta 1990, kar kaže na to, da večina dokumentacije ne izvira iz raziskav, temveč iz praktičnega dela.

10. Postavljena je na konec projekta razvoja poslovnega sistema (Morris, 2006)

Ponavadi je migracija podatkov postavljena bolj proti koncu projekta razvoja poslovnega sistema, kar ponavadi pomeni tudi, da je planiran strošek zanjo že precej okrnjen.

11. Podcenjena obsežnost in poenostavljene metode (Shepherd, 1999; Lashley, 2006; Lin, 2008)

Ker se velikokrat privzame, da je migracija podatkov bolj kopiranje podatkov z izvora na ciljni sistem, se ne predvidi ljudi z dovolj ekspertnega znanja in se podceni kompleksnost njihovega dela, kar pa lahko pripelje do podaljšanja projekta. To se lahko zgodi tudi:

- ko razvoj ciljne aplikacije zamuja;
- nimamo kompetentnih ljudi, ki bi razumeli podatke na izvornem sistemu, ali ne želijo sodelovati;
- pomanjkanje ljudi s kompetencami.

12. Skromen nabor orodij za podporo migraciji podatkov (Carreira in Galhardas, 2004)

Ponavadi je izdelana rešitev prilagojena za trenutni projekt in se jo redko ponovno uporabi. Orodja, ki bi podprla vse procese migracije podatkov, so redka. Obstajajo pa orodja, ki pomagajo v določenem procesu.

13. Potrebno je rezumevanje podatkov (Wu in ostali, 1997, Shepherd, 1999; Morris, 2006)

Da bi bila migracija uspešna, moramo razumeti podatke same ter tudi podatkovni model na izvoru in na ciljnem sistemu.

14. Popolna strategija (Morris, 2006)

Projekt migracije podatkov mora imeti definirano strategijo za dostop, implementacijo, verifikacijo in validacijo podatkov.

15. Sodelovanje (Morris, 2006; Russom, 2006; Wagner in Wellhausen, 2011)

Uspeh projekta je odvisen od sodelovanja in izmenjave informacij med poslovnimi analitiki in skupino, ki opravlja migracijo podatkov. Velik problem pri takem projektu je zamujanje informacij s poslovne strani.

Poglavje 3

Metode migracije podatkov

Prvi poskusi migracije so bili intuitivni. Z izkušnjami, ki so bile pridobljene pri različnih projektih migracije podatkov, pa so se oblikovale metode in dobre prakse, ki marsikomu tudi danes, še posebej pa začetnikom, olajšajo delo in prihranijo čas.

Z analizo obstoječe literature je nastal spodnji seznam možnih metod migracije podatkov. Razdeljene so na dva vidik, tehničnega in vodstvenega [7]. Tehnični vidik se nanaša na implementacijo tehnične rešitve, medtem ko se vodstveni vidik nanaša na migracijo podatkov na procesnem nivoju – vodenju različnih faz in aktivnosti migracije podatkov.

Tehnični vidik:

- Pretvorba sheme (angl. *Schema Conversion*),
- Sprememba aplikacije (angl. *Program Conversion*),
- Pristop meta modeliranja (angl. *Meta-Modeling*),
- ETL (angl. *Extract, Transform and Load*),
- Modelno usmerjena migracija podatkov (angl. *Model Driven Data Migration*),
- Avtomatizirana migracija podatkov (angl. *Automated Data Migration*).

Vodstveni vidik:

- Metoda metulja (angl. *The Butterfly Method*),
- Arhitekturno usmerjena prenova (angl. *Architecture driven Modernization*),
- Metoda The Data Warehouse Institutea (angl. *The Data Warehouse Institute Method*)
- Praktična migracija podatkov (angl. *Practical Data Migration*)
- Procesni model (angl. *Process Model*)
- Metoda migracije podatkov za podatkovno intenzivno produktno programsko opremo (angl. *Software Product Data Migration Method*)

3.1 Tehnični vidik

3.1.1 Pretvorba sheme (angl. *Schema Conversion*)

Migracija na novo platformo in tehnologijo je zahtevna in draga. Henrard in ostali [5] predlagajo različne strategije postopne migracije, ki združujejo tehnike na nivoju podatkovne baze in na nivoju programa, kar je manj tvegano.

Na nivoju podatkovne baze imamo pretvorbo fizične sheme in pretvorbo konceptualne sheme. Pri prvi simuliramo fizično shemo starega sistema, pri čemer se koncepti ne spreminjajo. Elemente iz starega sistema pretvorimo v najbližji konstrukt v novem sistemu (eno datoteko pretvorimo v eno tabelo). Ta rešitev je sicer poceni, vendar je kvaliteta podatkov v novem sistemu nizka, tudi zato, ker podatki niso pomensko interpretirani. Pri drugi strategiji, kjer je kvaliteta podatkov dobra, podatkovna baza dokumentirana, vendar je dražja, pa razvijamo podatkovno bazo novega sistema iz konceptualne sheme, podatki pa ohranijo pomen.

3.1.2 Sprememba aplikacije (angl. *Program Conversion*)

Henrard in ostali (2002) [5] omenjajo tri tipe strategij:

- Novomigriran ciljni podatkovni model je ovit na programskem nivoju in je vmesnik za staro aplikacijo, ki omili razliko pri klicih na izvornem in ciljnem podatkovnem nivoju.
- Prepis programskih poizvedb na podatkovnem nivoju.
- Prenovitev aplikacije, ki je lahko kompleksna in za katero je potrebno podrobno znanje o poslovni logiki sistema.

Warren in Ransom (2002) [7, 12] priporočata renesančno metodo (angl. *The Renaissance method*), ki s preoblikovanjem starega sistema v razvijajoč se sistem in uspešno izpeljanimi ponavljajočimi se izboljšavami omogoča, da je sistem usklajen s strategijo podjetja in omogoča nižje stroške in večjo uporabljivost sistema.

3.1.3 Pristop meta modeliranja (angl. *Meta-Modeling*)

Jeusfeld in Johnen (1995) [7, 6] sta se s tem pristopom poskušala izogniti odvisnosti od izvirnega in ciljnega podatkovnega modela tako, da sta vpeljala model metapodatkov podatkovnega modela, kjer so določene preslikave med izvornimi in ciljnim podatki.

Podobnega pristopa so se poslužili Maatuk in ostali (2008) [7], ki so s kanoničnim podatkovnim modelom (angl. *Canonical Data Model*) predstavili model izvornih metapodatkov podatkovnega modela - razširili podatkovno shemo izvornih podatkov - in tako poenostavili preslikavo ter s tem migracijo podatkov na kompleksne ciljne objekte.

Jahnke in Wadsack (1999) [7] pa sta predlagala dve fazi – določitev logične sheme s pomočjo analize izvirne podatkovne baze in preoblikovanje logične

sheme v konceptualno shemo, ki služi kot osnova za preslikavo med izvornimi in ciljinimi podatki.

3.1.4 ETL (angl. *Extract, Transform, Load*)

Haller (2009) [4] je na podlagi svojih izkušenj predlagal generično arhitekturo za migracijo podatkov, ki temelji na ETL-u (izvoz, preoblikovanje in vnos podatkov).

Za fazo izvoza podatkov moramo najprej vedeti, v katerih podatkovnih skladiščih se potrebni podatki nahajajo. Količino podatkov lahko zmanjšamo na željeno s filtriranjem, ki je lahko:

- na nivoju atributa,
- s pomočjo tabele za izbiranje,
- s funkcijami za agregacijo na več kot eni vrstici.

V fazi preoblikovanja restrukturiramo podatke z izvora, da ustrezajo shemi podatkov na ciljnem sistemu. Lahko imamo tri različne vzorce:

- Vzorec enostavnega prenosa atributov, kjer je isti atribut shranjen v različnih tabelah na izvoru in cilju.
- Vzorec razširitve, kjer imata izvorni in ciljni sistem podobne attribute, vendar se izvorni podatki uporabljajo na večih različnih mestih v izvornem sistemu.
- Vzorec zožanja, kjer je na izvornem sistemu več instanc istega atributa, na ciljnem pa manj.

Domenske vrednosti se v tej fazi preslikajo v ciljne domenske vrednosti z uporabo preslikovalnih tabel, v primeru zahtevnih preoblikovanj pa se lahko uporabijo preslikovalne funkcije.

Sledi faza vnosa podatkov, ki se lahko naredi na tri načine:

- Neposreden prenos podatkov z izvirnega sistema v tabele na ciljnem sistemu.
- Preko programskega vmesnika (API-ja), ki uporablja vmesno podatkovno skladišče za nalaganje podatkov, od koder podatke prebere programski vmesnik in jih zapiše v tabele na ciljnem sistemu.
- Preko programskega vmesnika (API-ja), ki temelji na delovnem toku. Podatki se prav tako najprej zapišejo v vmesno podatkovno skladišče, od koder se s pomočjo funkcij delovnega toka, za vsak objekt zapišejo v tabele na ciljnem sistemu. Ta način je podoben načinu, ki ga uporablja aplikacija na ciljnem sistemu za vnos novega objekta.

Sledi še testiranje s testnimi primeri in tehnična avtomatska verifikacija, če so bili vsi objekti med migracijo podatkov preneseni. Lahko se preveri pravilnost tudi na nivoju samih atributov, največkrat s pomočjo zgoščevalne funkcije (angl. *hashing*).

ETL je najbolj uporabljena metoda pri migriranju podatkov.

3.1.5 Modelno usmerjena migracija podatkov (angl. *Model Driven Data Migration*)

Po mnenju Bordbara in ostalih (2005) [7] ter Fleureya in ostalih (2007) [1] je migracija podatkov tesno povezana z razvojem ciljnega sistema, ki je tudi modelno usmerjen. Prinaša hitrejši razvoj in robustnost sistema, saj je proces avtomatiziran in je delo programerja potrebno samo, če moramo razumeti pomen podatkov. Najprej se naredi celovit model aplikacijske programske kode. Z obratnim inženirstvom (s transformacijo modela) se iz modela programske kode naredi model, ki je neodvisen od platforme. S tem korakom se povzame visokoravenski pogled na model programske kode. V naslednjem koraku se od platforme neodvisen model transformira v meta model programske kode, ki ustreza platformi ciljnega sistema. Na koncu se razvije še koda nove aplikacije, ki temelji na modelu, ki je specifičen za ciljno platformo.

3.1.6 Avtomatizirana migracija podatkov (angl. *Automated Data Migration*)

Avtomatizirana migracija podatkov je snovana na metodi Modelno usmerjena migracija podatkov. Avtomatizacija migracije se lahko doseže z uporabo repozitorija pravil in preslikav podatkovnih tipov med izvornim in ciljnim sistemom. Aboulsamh, Crichton, Davies in Welch (2010) [7] menijo, da sta prednosti zmanjšanje stroškov in število napak zaradi avtomatizacije (avtomatsko generiranje zaporedja transformacij podatkov).

Dobre in Marin (2011) [7] celo predlagata celostno arhitekturo za avtomatsko migracijo podatkov. Sestavljena je iz treh specializiranih modulov, ki med sabo sodelujejo in s tem vzdržujejo neprestan tok podatkov:

- Nivo dostopa do podatkov (angl. *Data Access Layer*)

Pridobiva informacije o podatkovni shemi izvirnega in ciljnega sistema ter nudi mehanizem za branje in vnašanje podatkov.

- Modul pretvorbe podatkov (angl. *Data Conversion Module*)

Izvaja pretvorbe podatkov, ki so potrebne za migracijo.

- Modul pravil preslikovanja (angl. *Rule Mapping Module*)

Služi za dodajanje novih in upravljanje že obstoječih pravil preslikovanja, ki so lahko tako implicitna kot eksplicitna. Ta modul ima od vseh treh največjo možnost avtomatizacije.

Module se nadzoruje preko uporabniškega vmesnika.

3.2 Vodstveni vidik

3.2.1 Metoda metulja (angl. *The Butterfly Method*)

Wu in ostali (1997) [13, 14] so razvili metodo metulja. Metoda je ena prvih metod, ki se ukvarja s procesom migracije podatkov. Metoda se osredotoča

na poznavanje starega in novega sistema ter na podatkovni vidik migracije. Pri njej izvorni in ciljni sistem ne delujeta istočasno, zato sinhronizacija med njima ni potrebna.

Ob začetku migracije podatkov se za izvorno podatkovno skladišče onemogoči pisanje (se zamrzne), dovoljeno je samo branje. Vse spremembe na izvornih podatkih so shranjene preko dodeljevalnika podatkovnega dostopa (angl. *Data-Access-Allocator*) v pomožna, začasna podatkovna skladišča. S pomočjo Chrysaliserja, ki je orodje razvito prav za ta namen in preoblikuje podatke, se podatki iz izvornega skladišča prenesejo v ciljno podatkovno skladišče. Transformacija je odvisna od sheme na izvornem in na ciljnem sistemu. Chrysaliser najprej prenese podatke iz zamrznjenega podatkovnega skladišča (TS0), vse spremembe na izvornih podatkih pa se zapišejo v začasno skladišče (TS1). Ko je prenos podatkov iz TS0 na ciljni sistem končan, se zamrzne TS1, ki se bo sedaj prenesel na ciljni sistem, Vse spremembe na izvornih podatkih se piše v naslednje začasno podatkovno skladišče (TS2). To se ponavlja toliko časa, dokler velikost trenutnega začasnega skladišča ni manjša ali enaka mejni vrednosti (angl. *Threshold Value*), ki jo ponavadi določi administrator. Podatki se tako prenašajo inkrementalno. Zadnje podatkovno skladišče je že tako majhno, da se lahko podatki v njem preoblikujejo in prenesejo tako hitro, da se lahko izvorni sistem odklopi in postane nedostopen tudi za branje, čas izpada sistema pa je majhen. Po prenosu zadnjih podatkov se ciljni sistem da v uporabo.

Metodo sestavlja 6 faz:

- **Faza 0:** Priprava na migracijo (angl. *Prepare for migration phase*)

Aktivnosti:

- Določijo se uporabniške zahteve.
- Določi se merilo za oceno uspešnosti migracije (kriteriji validacije).
- Določi se ciljna arhitektura.

- Določi se strojna oprema.
- **Faza 1:** Razumevanje semantike izvirnega sistema in razvoj ciljne sheme podatkov (angl. *Understand of semantics of legacy system and develop the target data schemas*)

Aktivnosti:

- Analizirajo se vmesniki, aplikacija in podatki izvirnega sistema.
 - Določi se redundanca pri vmesnikih in funkcionalnosti aplikacije.
 - Določi se obseg podatkov, ki se bo migriral.
 - Opredeli in razišče se interakcija z drugimi sistemi.
 - Implementira se dodeljevalnik podatkovnega dostopa (angl. *Data-Access-Allocator*).
 - Implementira se ciljna shema podatkov.
 - Določi se pravila za preslikavo med izvirno in ciljno shemo podatkov.
- **Faza 2:** Grajenje vzorčnega podatkovnega skladišča, ki temelji na ciljnih vzorčnih podatkih (angl. *Building up a Sample Datastore, based upon the Target SampleData, in the target system*)

Aktivnosti:

- Pripravijo se testni podatki za ciljni sistem.
 - Implementacija Chrysaliserja.
- **Faza 3:** Inkrementalno migriranje vseh komponent izvirnega sistema na ciljno arhitekturo razen podatkov (angl. *Incrementally migrate all the components (except for data) of the legacy system to the target architecture*),

Aktivnosti:

- Migracija vmesnikov izvirnega sistema.
 - Migracija aplikacijske funkcionalnosti izvirnega sistema.
 - Migracija komponent izvirnega sistema, ki bodo ponovno uporabljene.
 - Integracija komponent in sistemov, ki so bili uporabljeni na izvirnem sistemu.
 - Verifikacija migriranih komponent in ciljnega sistema.
 - Validacija migriranih komponent in ciljnega sistema (ali so v skladu z uporabniškimi zahtevami).
 - Usposabljanje uporabnikov na ciljnem sistemu.
- **Faza 4:** Postopna migracija izvornih podatkov na ciljni sistem in usposabljanje končnih uporabnikov na novem sistemu (angl. *Gradually migrate the legacy data into the target system and train users in target system*),

Aktivnosti:

- Vključitev dodeljevalnika podatkovnega dostopa (angl. *Data-Access-Allocator*) v izvorni sistem.
- Ustvari se začasno podatkovno skladišče TS1, TS0 se spremeni, da je možno samo branje iz njega.
- S pomočjo Chrysaliserja se vsi podatki iz TS0 prenesejo v ciljni sistem. Medtem se ves dostop do izvornih podatkov preusmeri (dodeljevalnika podatkovnega dostopa) in rezultat pretvorbe podatkov se shrani v začasno podatkovno skladišče TS1.
- Ustvari se novo začasno podatkovno skladišče TS2, za TS1 se nastavi, da je možno samo branje iz njega.
- Vsi podatki se migrirajo iz TS1 v ciljni sistem. Rezultat pretvorbe podatkov se shrani v TS2.
- To se ponavlja, dokler pogoj za konec migracije ni dosežen.
- Celoten izvorni sistem se zamrzne, zmigrirajo se še zadnji podatki.

- Usposabljanje uporabnikov na ciljnem sistemu.
- **Faza 5:** Preklop na novi sistem (angl. *Cut-over to the completed target system*).

Aktivnosti:

- Preklop na novi sistem, odklop starega sistema.

Lastnosti metode:

- Poudarek je na testiranju, ki je jasno določeno. Iz izkušenj je razvidno, da se za testiranje lahko porabi tudi do 80% časa na projektu za prenovo sistema. Ciljna aplikacija se lahko celovito in učinkovito stestira na podlagi podatkov, ki so shranjeni v vzorčnem podatkovnem skladišču.
- Je prilagodljiva, saj se ne naslanja na katerokoli orodje za migracijo podatkov (z izjemo Chrysaliserja in dodeljevalnika podatkovnega dostopa). Tako se lahko uporabi katerokoli orodje kot pomoč pri migraciji.
- Trajanje celotne migracije podatkov se lahko oceni. Potreben čas se lahko enostavno izmeri na podlagi količine podatkov v izvornem sistemu in hitrosti Chrysaliserja in dodeljevalnika podatkovnega dostopa. Realistična ocena nam je lahko v veliko pomoč pri planiranju, kar omogoča tudi večjo verjetnost uspešne migracije.
- Posredovanje med izvornim in ciljnim sistemom je minimalno. Transformacija se izvaja enosmerno med delovanjem sistema, ostalo lahko opravimo, ko je sistem odklopljen.
- Prekinitev izvornega sistema je minimalna. Izvorni sistem bo deloval vse do migracije zadnjega začasnega podatkovnega skladišča, kjer pa bo čas izpada sistema majhen.
- Podpira paralelno izvajanje aktivnosti.

Uspešnost metode je odvisna od:

- Razumevanja izvirnega in ciljnega sistema.
- Točnosti in uporabnosti vzorčnih podatkov.
- Hitrosti delovanja Chrysaliserja.
- Zmogljivosti dodeljevalnika podatkovnega dostopa (angl. *Data-Access-Allocator*).

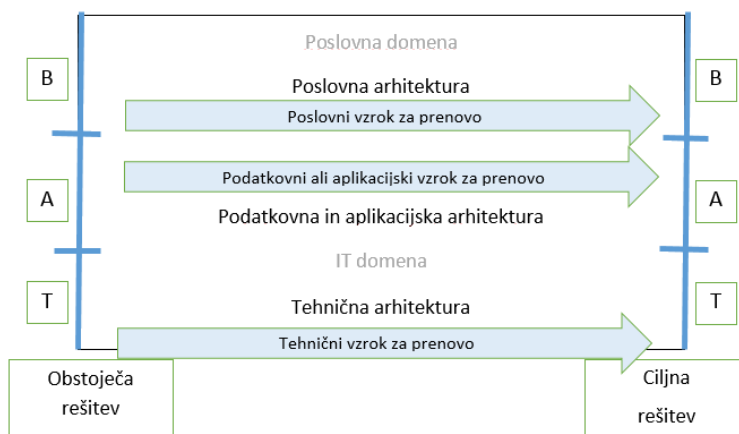
3.2.2 Arhitekturno usmerjena prenova (angl. *Architecture driven Modernization*)

Arhitekturno usmerjena prenova obravnava prenovo informacijskega sistema na treh nivojih:

- Tehničnem,
- podatkovnem in aplikacijskem,
- poslovnem.

Opazen je poudarek na povezanosti med nivoji, ki morajo biti vedno usklajeni med sabo. Največkrat je vzrok za prenovo tehničen, saj strojna oprema hitro zastari. Aplikacijski in podatkovni nivo je vzrok za prenovo, ko aplikacija ne zadosti več poslovnim potrebam ali ko je podatkovni model zastarel. Če se spremeni poslovni model (poslovna semantika, pravila ali procesi), je vzrok za prenovo poslovni. Domene in nivoji so razvidni na Sliki 3.1.

Prenova se je nekoč osredotočala na tehnični nivo. Danes pa je postalo pomembno zaradi potrebe po prenovi ali spremembi poslovnih pravil ali procesa in s tem ponavadi tudi potrebe po prenovi na podatkovnem in aplikacijskem nivoju, da se prenova izvaja na vseh treh nivojih. Povezava med nivoji naj bi bila čim večja, opisuje pa jo model podkve (angl. *Horseshoe Model*), ki sta ga predlagala Khusidman in Ulrich (2007). [8]



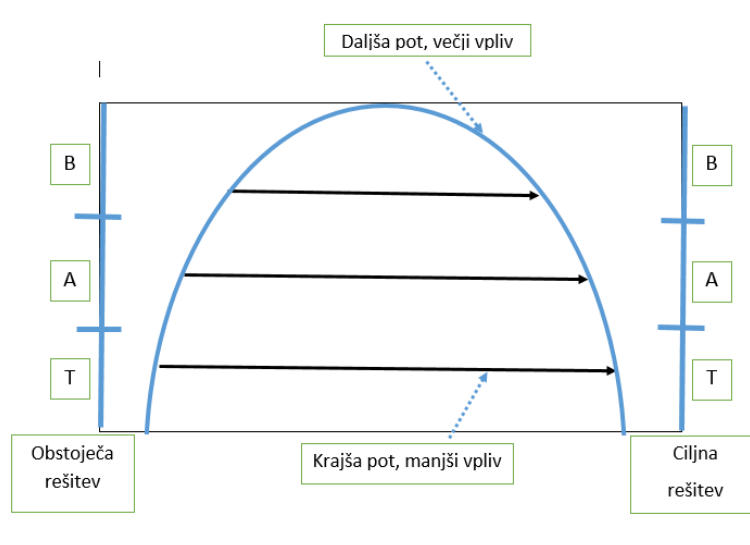
Slika 3.1: Domene in nivoji prenove.

Model podkve vključuje enega ali več nivojev. Vsaka komponenta prenove ima lahko svojo krivuljo prenove iz obstoječega v ciljno stanje.

Pot predstavlja, kako se pridobi znanje o trenutnem stanju, kako se to znanje poveča in ponovno uporabi pri nadgradnji na novo stanje. Iz obstoječega stanja lahko pridemo v željeno stanje po krajši ali daljši poti. Odločitev, po kateri poti bo prenova izpeljana, mora biti dobro premišljena, saj ni nujno, da krajša pot zadovolji vse zahteve, po drugi strani pa daljša pot morda v danem trenutku ni izvedljiva. Lahko se zgodi, da se za prenovo sistema uporabi inkrementalni pristop, kjer se na začetku izbere krajšo pot, kasneje pa se naredi prenova po daljši. Model podkve je viden na Sliki 3.2.

Pri vsaki poti prenove, ne glede na nivo vpliva na arhitekturo, imamo tri elemente:

- Pridobitev znanja o obstoječem stanju.
- Definicija arhitekture na ciljnem sistemu.
- Koraki preslikave med obstoječim stanjem in ciljnim stanjem.



Slika 3.2: Model podkve.

Poskrbeti moramo za usklajenost arhitekturnih nivojev tako vertikalno (od poslovnega nivoja do fizičnega) kot horizontalno (od obstoječega do ciljnega stanja). Model podkve za strategijo prenove poslovnega sistema je kombinacija poti prenove (preslikave v vertikalni in horizontalni smeri), ki temeljijo na poslovnih in IT zahtevah.

3.2.3 Metoda The Data Warehouse Institutea (angl. *The Data Warehouse Institute Method*)

Metoda, ki jo je opisal Russom (2006) [7], temelji na iterativnem pristopu razvoja migracije podatkov. Predstavlja dobre prakse pri migraciji podatkov ter model razvoja in model izvedbe migracije podatkov, ki so bili uporabljeni za The Data Warehouse Institute.

Model razvoja migracije podatkov sestavlja skupaj s pripravljalno fazo šest faz:

- Pripravljalna faza (pred načrtovanjem rešitve)

- Zbiranje zahtev.
- Izdelava podrobnega projektnega plana in časovnice (angl. *timeline*).
- Profiliranje podatkov, ki naj bi se izvajalo skupaj s poslovnimi vodji in eksperti, ki poznajo domeno.
- Faza načrtovanja rešitve
 - Se ukvarja z razbitjem migracije podatkov na ločene naloge glede na njihovo odvisnost.
- Faza modeliranja podatkov
 - Cilj faze je izdelava ciljne podatkovne baze, ki se lahko izvaja tudi postopno z vmesnimi modeli.
 - V tej fazi se lahko izvaja tudi izboljšava na nivoju metapodatkov.
- Faza preslikovanja podatkov
 - Določi se preslikava podatkov ter pravila preoblikovanja.
 - Proces je iterativen, tako se pravila preoblikovanja in vrstni red migracije podatkov pokaže po večih ciklih definiranja, razvoja in testiranja.
 - Zaradi kompleksnosti te faze je priporočljivo, da se preslikave ne določajo ročno, temveč se uporabi orodje za pomoč pri migraciji podatkov z uporabniškim grafičnim vmesnikom.
- Faza razvoja rešitve
 - Razvije se rešitev za migracijo podatkov, ki učinkovito migrira podatke z izvirnega sistema na ciljni sistem z uporabo pravil preslikav in preoblikovanja.
- Faza testiranja
 - Potrebno je določiti reprezentativen vzorec podatkov na katerem se bo izvajalo testiranje.

Razvoj in testiranje tvorita cikel, s katerim se rešitev iterativno izboljšuje.

Model izvedbe migracije podatkov sestavlja pet faz:

- Faza izvedbe
 - Izvede se migracija podatkov.
- Administrativna predaja sistema
 - Razvojna skupina za migracijo preda odgovornost za podatke IT oddelku.
- Faza sinhronizacije
 - Ta faza je potrebna samo, če stara in prenovljena aplikacija sočasno delujeta in se morajo podatki zato med sistemoma sinhronizirati.
- Faza spremljanja
 - Pomembna faza, kjer končni uporabniki ocenijo uspešnost migracije podatkov.
 - Analizirajo se razširljivost, izpadi in podatkovne napake sistema.
- Faza umika starega sistema
 - Umik stare aplikacije in podatkov.
 - Pred umikom je potrebno preveriti, če je bila migracija podatkov uspešno izvedena, sicer vrnitev na prejšnje stanje ni več mogoče.

3.2.4 Praktična migracija podatkov (angl. *Practical Data Migration*)

Morris (2006) [11] je naredil velik korak za migracijo s popisom izkušenj praktičnih izvedb migracije podatkov na različnih projektih. Izkušnje so

predstavljene z vidika zunanje osebe, kateri je bila zaupana naloga migracije za naročnika. Metodo za uspešno izvedbo migracije, ki jo je definiral, je izpopolnil ponovno leta 2012, opredelil pa je tudi koncepte in zakonitosti migracije podatkov. Skupnosti, ki se ukvarja z migracijo podatkov, so znana štiri zlata pravila, ki jih zagovarja Morris:

- Migracija podatkov je poslovni, ne tehnični problem.
 - Čeprav je migracija podatkov tehnične narave, pa podatki, ki s poslovnega vidika ne nosijo informacije, niso uporabni. Zato mora biti lastnik podatkov nekdo, ki pozna poslovni del sistema in lahko določi vrednost podatkov.
- Vodje poslovnih področij vedo najbolje.
 - Vodje poslovnih področij najbolje poznajo delovanje sistema in podatke, ki so pomembni, zato znajo tudi opredeliti, kateri podatki in v kakšni obliki se bodo migrirali.
- Podjetja ne potrebujejo, ne želijo in ne bodo plačala za popolno kvaliteto podatkov.
 - Ker je proces čiščenja podatkov zahteven in dolgotrajen, je potrebno z lastnikom podatkov in vodjami poslovnih področij določiti nivo čiščenja podatkov.
- Če se ne da prešteti, ne šteje.
 - Da bi lahko določili napredek in kvaliteto migracije, je potrebno določiti merljive parametre.

Za uspešno migracijo podatkov se morajo projektni vodje zavedati štirih pomembnih konceptov glede podatkov obstoječega sistema in strategij migracije, ki so osnova za najpomembnejše končne izdelke migracije.

- Izvorno podatkovno skladišče (angl. *Legacy Data Store*)
 - Kakršenkoli repozitorij, ki vsebuje podatke, relevantne za migracijo. Podatkovna skladišča je potrebno najti, analizirati in popisati.
- Relevantne poslovne domene (angl. *Key Business Data Areas*)
 - Sestavljajo jih različna izvorna podatkovna skladišča in poslovne funkcije, ki hranijo podatke o določeni entiteti konceptualnega modela trenutnega sistema. Kot izvorno podatkovno skladišče je potrebno tudi poslovne domene analizirati in popisati ter na podlagi analize določiti strategijo migracije.
- Pravilo za umik starega sistema (angl. *System Retirement Policy*)
 - Določa postopek za umik izvornih podatkovnih skladišč po uspešno opravljeni migraciji.
- Pravila za prenos podatkov (angl. *Data Transitional Rules*)
 - V primeru, ko star sistem deluje med procesom migracije podatkov, je potrebno zagotoviti, da se tudi novonastali podatki prenesejo v nov sistem.

Avtor metode Praktična migracija podatkov poudarja, da mora biti analiza vrzeli med starim in novim sistemom natančno opravljena, saj lahko le tako pravila za preslikavo podatkov dobro in pravilno določimo.

Zahteve, ki morajo biti določene, so še:

- Količina podatkov, ki jo je potrebno migrirati.
- Čas, ki je potreben za migracijo te količine podatkov.
- Zaporedje izvedbe procesa migracije.

- Strojna in programska oprema, ki se uporablja za izvirno podatkovno skladišče.
- Časovno okno v katerem lahko izvedemo migracijo (čas, ko se sistem najmanj uporablja).
- Strategija za povrnitev sistema v prvotno stanje, če pride do problemov.
- Strategija za problem enosmerne ceste, ko se izvorni podatki med migracijo transformirajo in originalne vrednosti ni mogoče več izslediti. Rešitev za ta problem je kloniranje izvirnega podatkovnega skladišča.

Izvedba migracije lahko poteka na tri načine:

- Migracija v enem koraku (angl. *Big Bang*)
 - Vsi podatki se migrirajo naenkrat. Ko je proces migracije podatkov končan, se izvirno podatkovno skladišče umakne iz uporabe.
- Paralelno delovanje (angl. *Chicken little*)
 - Med izvajanjem migracije so izvirna podatkovna skladišča še v uporabi. Šele ko se migracija konča in so vodje poslovnih področij zadovoljni z rezultatom migracije, se izvirna podatkovna skladišča lahko umakne.
- Dostava po fazah (angl. *Butterfly*)
 - Migracijo se razdeli na več faz, vsaka faza pa je neka zaključena celota, ki omogoči delovanje nove ali prenovljene funkcionalnosti sistema.

Avtor zagovarja ločitev projekta migracije od krovnega projekta prenove sistema. Njegovi razlogi za to so:

- Migracija podatkov ima specifične izdelke, kot so:

- Pravila za kvaliteto podatkov,
 - pravila za umik starega sistema,
 - pravila za prenos podatkov,
 - analiza deležnikov podatkov.
- Skupina za migracijo podatkov mora komunicirati tako s poslovno kot tehnično stranjo.
- Poslovni analitiki za migracijo podatkov morajo imeti specifično, specializirano znanje.
- Struktura projekta, v primerjavi s klasičnim projektom razvoja, se razlikuje.

Projekt migracije podatkov ima štiri faze:

- Faza 0: Faza inicializacije
 - V tej fazi se postavi osnova za projekt migracije podatkov. Identificira se vse vpletene pri procesu (vse deležnike pri manipulaciji podatkov in eksperte na poslovnem področju), določi se časovne in stroškovne omejitve in mejnike za dostavo delnih in končnih izdelkov ter kdo je zadolžen za njihovo dostavo. Zabeleži se vse lastnike izvornih podatkovnih skladišč ter se dokumentira izvorna podatkovna skladišča ter relevantne poslovne domene. Določi se strategijo za način izvedbe migracije podatkov ter strategijo testiranja.
- Faza 1: Priprava podatkov
 - V fazi priprave podatkov se poskuša rešiti probleme, ki smo jih zaznali pri analiziranju izvornih podatkovnih skladišč. Določi se, katera izmed izvornih podatkovnih skladišč se bodo migrirala, se bodo ohranila nespremenjena ali bodo odstranjena. V tej fazi se določi tudi pravila za kvaliteto podatkov ter modele relevantnih poslovnih domen. Postavi se inicialna oblika strategije za način umika starega sistema iz uporabe.

- Faza 2: Priprava podatkov
 - Rezultat faze so očiščena izvorna podatkovna skladišča, ki so v skladu s konceptualnim modelom izvirnega sistema. Za pomoč pri čiščenju se uporabljajo prehodna podatkovna skladišča. Analizira se tudi vrzel med starim in novim sistemom. Kjer podatki manjkajo, pa so obvezni, se določijo privzete vrednosti. V primeru, če bosta tako stari kot novi sistem delovala istočasno, se določi pravila za sinhronizacijo podatkov. Pomembni izdelki po koncu faze 1 in 2 so med drugim pravila za kvaliteto podatkov, pravilo za umik starega sistema iz uporabe in konceptualni model entitet in povezav.
- Faza 3: Razvoj in testiranje
 - Čeprav aktivnosti te faze ustrezajo aktivnostim na projektu razvoja, pa vsebujejo določene posebnosti, ki veljajo le za migracijo podatkov. Za izdelavo fizičnega načrta za proces migracije podatkov se uporabijo:
 - * Definicija za izvoz podatkov iz starega sistema, preoblikovanje podatkov in vnos podatkov v novi sistem (definicija ETL-a),
 - * pravila za sinhronizacijo podatkov,
 - * pravila za umik starega sistema iz uporabe.

Načrt vsebuje časovno premico s posameznimi aktivnostmi, tehnične specifikacije, korake za povrnitev sistema v prvotno stanje, če pride do problemov, ter nefunkcionalne specifikacije. V tej fazi se določi strategija testiranja, razvije se orodje za izvedbo migracije podatkov ter se izvede migracija. Po vsaki testno izvedeni migraciji podatkov se preveri rezultat in se po potrebi popravi ali nadgradi orodje za migracijo ter ponovno prečisti podatke.

Predvsem med fazo priprave podatkov je iteracija pogosta. Med samim projektom pa se mora velikokrat spreminjati transformacijo, preslikavo in sam način migriranja podatkov, še posebej, če se ciljni sistem še razvija. Pogosto

se zgodi, da nekateri podatki ne ustrezajo novemu sistemu ali bi celo ogrozili njegovo delovanje, zato je potrebno z vodjami poslovnih področij analizirati moteče podatke in določiti vrednosti, ki so sprejemljive za nov sistem in so s poslovnega vidika še vedno uporabne. Razdelitev procesa migracije na več faz omogoča, da se po koncu vsake faze preveri plan za naslednjo fazo in stanje. Tako lahko plan prilagodimo, če je to potrebno.

3.2.5 Procesni model (angl. *Process Model*)

Matthes, Schultz in Haller (2011) [2, 3] so predstavili celosten procesni model migracije podatkov velikega obsega. Razdeljen je na štiri stopnje:

- Stopnja 1: Inicializacija (postavitev organizacije in infrastrukture za migracijo)
 - Faza 1: Javni razpis in ponudba (angl. *Call for tender and bidding*)
 - * Kdo bo izvedel migracijo (notranji oddelek ali zunanje izvajanje)
Razlog za izbor zunanjega izvajalca so trije:
 - ◊ Projekt migracije podatkov zahteva dodaten čas in trud poleg vseh ostalih nalog, ki jih opravlja notranji IT oddelek. Ponavadi je vložek podcenjen.
 - ◊ Znanje za take projekte se gradi počasi, trud potreben pa je velik. Večinoma se ne izplača imeti oddelka samo za migracijo podatkov.
 - ◊ Migracija podatkov je redek dogodek v okviru neke poslovne organizacije, zato ni veliko priložnosti za pridobivanje znanja na takih projektih.
 - Faza 2: Strategija in predhodna analiza (angl. *Strategy and pre-analysis*)
 - * Določitev obsega migracije (kateri podatki in v kakšni količini se bodo prenesli, identificira se poslovne koncepte, ki se jih preslika v

tabele in kolone na tehničnem nivoju; poslovni koncepti so lahko na poslovnem nivoju enaki, razlikujejo pa se v izvedbi na tehničnem nivoju, lahko pa se razlikujejo tako na poslovnem kot tudi na tehničnem nivoju.).

- * Določitev strategije za migracijo (migracija v enem koraku ali inkrementalno).
- * Izdelava projektnega plana.
- * Popisati je potrebno izzive in pripraviti rešitve (velika količina podatkov, migriranje podatkov v ciljni sistem, ki že vsebuje podatke, slaba kvaliteta podatkov, itd.).
- Faza 3: Postavitev platforme (angl. *Platform setup*)
 - * Definiranje tehnične infrastrukture, ki jo bomo uporabljali pri migraciji.
- Stopnja 2: Razvoj (programa za migriranje podatkov)
 - Faza 1: Izvoz podatkov (angl. *Data unloading*)
 - * Izvoženi podatki so trenutni posnetek izvornih podatkov, ki so že filtrirani glede na odločitev o obsegu podatkov v prejšnji stopnji. Kasnejši izvozi podatkov zajemajo samo podatke, ki so bili na novo dodani ali so bili spremenjeni od prešnjega trenutnega posnetka.
 - * Z validacijo izvoženih podatkov zagotovimo, da so podatki za migracijo enaki tistim na izvornem podatkovnem skladišču.
 - Faza 2: Analiza strukture podatkov in podatkov na izvornih in ciljnih podatkovnih skladiščih (angl. *Structure and data analysis for source and target*)
 - * Pomembna faza, saj si z dobrim znanjem o vsebini podatkov in načinu, kako so podatki shranjeni v podatkovnih skladiščih (tako na izvornem sistemu kot tudi na ciljnim sistemu), olajšamo delo pri ostalih stopnjah in fazah migracije. Faza je sicer dolgotrajna in

zahtevna, analiza izvornih podatkovnih struktur pa se lahko izvaja vzporedno z analizo ciljnih podatkovnih struktur.

- Faza 3: Čiščenje izvornih podatkov (angl. *Source data cleansing*)
 - * S čiščenjem izboljšamo kvaliteto podatkov. Priporočljivo je, da se čiščenje izvaja pred preoblikovanjem, saj lahko tako pospešimo preslikovanje in preoblikovanje podatkov, ker imamo manj različnih vrednosti za obravnavo. Možno pa je čiščenje opraviti tudi na ciljnim podatkovnem skladišču ali v fazi preoblikovanja.
- Faza 4: Preoblikovanje podatkov (angl. *Data transformation*)
 - * V tej fazi inkrementalno izboljšujemo pravila in logiko za migracijo podatkov. Sestavljeno je iz poslovnih in tehničnih pravil za preoblikovanje podatkov, ki povedo, kakšne so povezave med izvornimi in ciljnim atributi. Poslovna pravila so manj formalen opis odvisnosti, tehnična pravila pa določajo preslikavo med izvornimi in ciljnim atributi. Možna pravila za preoblikovanje so:
 - ◇ Neposredna preslikava, kjer se izvorni atribut preslika ena na ena v ciljni atribut.
 - ◇ Ciljni atribut ima za vrednost neko vnaprej definirano konstanto
 - ◇ Ciljni atribut ima za vrednost neko vnaprej definirano konstanto, če ni določeno drugo pravilo.
 - ◇ Domenska preslikava, kjer se vrednosti izvirnega atributa neposredno preslikajo v vrednosti, ki jih lahko zavzame ciljni atribut.
 - ◇ Preslikovalna tabela, kjer se pogleda v tabelo in se za vrednosti izvirnega atributa izbere pripadajoča vrednost za ciljni atribut.
 - ◇ Preslikovalna funkcija, kjer se vrednost za ciljni atribut določi s funkcijo, parametri pa so en ali več izvornih atributov.
 - ◇ Generirana vrednost, kjer je vrednost za ciljni atribut določena ne glede na izvorni atribut.
- Stopnja 3: Testiranje (validacija pravilnosti, stabilnosti in izvedbe migracije),

- Testiranje izvajanja migracije.
- Validacija podatkov:
 - * Testiranje celovitosti,
 - * Testiranje konsistentnosti podatkovnih tipov,
 - * Testiranje konsistentnosti podatkov,
 - * Testiranje preko uporabniškega vmesnika ciljne (nove) aplikacije.

Na koncu testiranja se pripravi poročilo, ki služi za naslednjo iteracijo, kjer se popravijo nepravilnosti med izvornimi in ciljnim podatki.

- Stopnja 4: Preklop na nov sistem
 - Faza 1: Migracija podatkov na nov sistem (angl. *Productive migration*)
 - * Potrebni pogoji za izvedbo migracije podatkov na nov sistem so uspešni testi in dokončana aplikacija na ciljnem sistemu. Preden se migracija izvede, se v primeru migracije v enem koraku izvorni sistem v določenem trenutku zamrzne, da ne bi nastajali novi podatki v izvornem sistemu. V primeru inkrementalne migracije to ni potrebno.
 - Faza 2: Dokončanje migracije (angl. *Finalizing*)
 - * Po izvedbi migracije podatkov se odgovornost za podatkovna skladišča preda lastniku podatkov oziroma njegovemu IT oddelku. Ko je sistem že v uporabi, se ga še naprej spremlja in se ugotavlja, ali so podatki celoviti, kakšna je njihova kvaliteta, znanje pa se lahko uporabi pri naslednji migraciji.
 - Faza 3: Čiščenje podatkov na ciljnem sistemu (angl. *Target data cleansing*)
 - * V primeru, da se je čiščenje podatkov na izvoru izpustilo ali pa se izkaže, da kvaliteta podatkov ni zadovoljiva, se izvede čiščenje po migraciji podatkov na ciljnem sistemu.

3.2.6 Metoda migracije podatkov za podatkovno intenzivno produktno programsko opremo (angl. *Software Product Data Migration Method*)

Čeprav je Morrisu uspelo zelo podrobno opisati projektni plan migracije podatkov, pa je Idu (2012) [7] mnenja, da se je preveč osredotočil na projektno planiranje, manjkajo pa podrobnosti glede upravljanja s spremembami in obvladovanja tveganj. Motila ga je tudi morebitna pomankljivost pri izvedbi migracije, saj Morris zagovarja, da je kvaliteta orodja za migracijo podatkov lahko le dovolj dobra, ni potrebno, da je optimalna. [11] Zato je Idu po raziskavi področja popisal seznam aktivnosti, ki po njegovem mnenju pomagajo pripeljati projekt migracije podatkov uspešno do konca v dogovorjenem časovnem in stroškovnem okviru.

Za razliko od ostalih avtorjev v svojih priporočilih ni upošteval samo primera, ko se migracija izvede enkratno, pogodbeno za naročnika, ob uvedbi novega sistema ali združevanju večih sistemov, temveč je upošteval tudi primer, ko imamo opravka s produktno programsko opremo. V takem primeru se migracija lahko izvaja vsakič, ko se sistem, ki sicer generira veliko podatkov, nadgradi, kar lahko pomeni tudi večjo spremembo podatkovne sheme. Večina avtorjev je predvidevala, da je ciljni sistem že razvit, izvajalec migracije podatkov pa lahko začne z aktivnostmi takoj, ko je faza testiranja ciljnega sistema končana. V primeru, ko je izvajalec migracije zadolžen tudi za razvoj ciljnega sistema, pa je potrebno faze prilagoditi in biti pozoren na odvisnosti med projektom migracije podatkov in projektom razvoja ciljnega sistema.

Idu je primerjal postopke do zdaj znanih metod migracije in obdržal aktivnosti, za katere je menil, da bi bile lahko uporabne pri vsaki migraciji podatkov. Ker je ena od zahtev, da se projekt migracije prilagaja krovnemu projektu razvoja novega sistema, je faze projekta razvoja programske opreme prilagodil fazam projekta migracije. Po metodi situacijske gradnje migracijske

metode je dodal aktivnosti, za katere je menil, da manjkajo ali da so premalo podrobne, ter na koncu faze in procese oblikoval v metodo. Metoda je sestavljena modularno, kar pomeni, da niso vse aktivnosti obvezne, temveč lahko neko aktivnost izpustimo, če bi bila njena izvedba nesmiselna. To velja predvsem v primeru, ko je projekt migracije manj kompleksen. Prav tako se lahko vrstni red aktivnosti in faz prilagodi potrebam, aktivnosti različnih faz pa lahko izvajamo tudi sočasno.

Poudaril je, da je sodelovanje vodij poslovnih področij, ki poznajo procese poslovanja, nujno, zato je tudi na koncu vsake faze dodal aktivnost, kjer se rezultat faz preveri in potrdi ustreznost z zahtevami in poslovnimi procesi. Metodo sestavlja pet faz, vsako pa sestavlja več aktivnosti:

- Inicializacija (angl. *Initialization*)
 - Določi vse deležnike podatkov (angl. *Identify Data Stakeholders*)
 - * Razumevanje, kdo vse vpliva na izvirne podatke in kdo je odvisen od njih.
 - Napravi predhodno poslovno analizo (angl. *Perform Business Pre-Analysis*),
 - * Analiza in identifikacija poslovnih konceptov in procesov, ki zahtevajo migracijo podatkov, ter določitev načina izvedbe migracije na poslovnem nivoju.
 - Napravi predhodno tehnično analizo (angl. *Perform Technical Pre-Analysis*),
 - * Analiza in identifikacija obsega migracije na tehničnem nivoju ter določitev načina izvedbe migracije na tem nivoju.
 - Razširi predhodno analizo razvoja produktne programske opreme (angl. *Extend Product Development Pre-Analysis*),

- * V primeru, ko imamo opravka z razvojem produktne programske opreme, se aktivnost predhodne analize na razvojnem projektu podaljša, da se analizira tudi vpliv na projekt migracije.
- Določi zahteve za migracijo in omejitve (angl. *Identify Requirements & Constraints*),
 - * Popiše se seznam funkcionalnih in tehničnih zahtev za migracijo ter omejitev.
- Izdelaj strategijo migracije podatkov (angl. *Create Data Migration Strategy*),
 - * Na podlagi prejšnje aktivnosti (Določi zahteve za migracijo in omejitve) se določi strategijo za celotni proces migracije podatkov. V primeru produktne programske opreme se strategija prilagodi strategiji razvoja produkta, čeprav sta še vedno neodvisni. Za izbor strategije je zadolžen vodja migracije podatkov (in vodja razvoja produkta).
- Oцени potrebni čas in stroške (angl. *Estimate Time & Cost*),
 - * Na podlagi aktivnosti predhodne analize se naredi prvi izračun vložkov potrebnih za projekt migracije.
- Izdelaj projektni plan razvoja rešitve za migracijo podatkov (angl. *Create Data Migration Development Project Plan*),
 - * Upoštevati je potrebno vse aktivnosti in vse udeležence na projektu.
- Izberi programsko orodje za migracijo podatkov (angl. *Choose Software Tools*),
 - * Razišče se možnost uporabe podpornih orodij za migracijo glede na popisane zahteve in omejitve ter, če katero od orodij ustreza, se ga uporabi v kasnejših fazah projekta.
- Vzpostavi okolje (angl. *Setup Data Migration Development Platform*)

- * Postavi se okolje, na katerem se lahko razvija programska rešitev za migracijo. Če se odločimo za uporabo podpornega orodja za pomoč pri migraciji, se ga prav tako namesti.
- Pridobi odobritev s strani vodij poslovnih področij (angl. *Get Business Approval*)
 - * S to aktivnostjo zagotovimo, da so vodje poslovnih področij vključeni v fazi inicializacije.
- Analiza (angl. *Analysis*)
 - Analiziraj izvorna podatkovna skladišča (angl. *Analyze Source Data Stores*)
 - * Identificirati je potrebno podatkovna skladišča, katerih podatki naj bi se migrirali. Podrobno je potrebno analizirati tako podatkovni model kot podatke same. V primeru, da imamo opravka s produktom, je znanje o izvornih podatkovnih skladiščih že prisotno in lahko zmanjšamo vložek na aktivnosti.
 - Analiziraj ciljna podatkovna skladišča (angl. *Analyze Target Data Stores*)
 - * Potrebno je podrobno analizirati podatkovno skladišče na ciljnem sistemu; podatkovni model ter same podatke, če ti že obstajajo v ciljnem sistemu. V primeru, da imamo opravka s produktom, je znanje o ciljnih podatkovnih skladiščih že prisotno in lahko zmanjšamo vložek na aktivnosti.
 - Analiziraj poslovne procese na izvornem sistemu (angl. *Analyze Source Business Processes*)
 - * Za dobro poznavanje poslovnih procesov na izvornem sistemu jih je potrebno podrobno preučiti. V primeru, da imamo opravka s produktom, je znanje že prisotno in lahko zmanjšamo vložek na aktivnosti.

- Analiziraj poslovne procese na ciljnem sistemu (angl. *Analyze Target Business Processes*)
 - * Da bi podatke pravilno prenesli in jih shranili v pravi obliki, moramo opraviti podrobno analizo poslovnih procesov na ciljnem sistemu. V primeru, da imamo opravka s produktom, je znanje že prisotno in lahko zmanjšamo vložek na aktivnosti.
- Razširi analizo izvirnega sistema (angl. *Extend Source Analysis*)
 - * Pri produkti programski opremi se analiza modela izvirnega sistema in poslovnih procesov trenutne verzije produkta razširi, da se naredi enako še za migracijo, da aktivnosti ni potrebno ponovno opraviti.
- Analiziraj razlike (angl. *Analyze Differences*)
 - * Pripraviti moramo podrobno analizo vrzeli med podatkovnim modelom starega in novega sistema, da bi tako imeli informacijo, kako definirati pravila za preslikavo.
- Določi pravila za kvaliteto podatkov (angl. *Define Data Quality Rules*)
 - * Določi se nivo kvalitete, ki jo morajo dosegati izvorni podatki, da so primerni za migracijo.
- Analiziraj integracije z drugimi sistemi (angl. *Analyze Interaction With Other Systems*)
 - * Rezultat aktivnosti je dokumentacija vseh sistemov, ki dostopajo do izvornih podatkovnih skladišč, ki hranijo podatke, ki naj bi se migrirali.
- Izdelaj strategijo za odklop starega sistema iz uporabe (angl. *Create System Retirement Policies*)
 - * Opisati moramo postopek, kako bomo odklopili izvorna podatkovna skladišča po tem, ko je bila migracija uspešno izvedena.
- Razširi analizo razvoja produkta (angl. *Extend Product Development Analysis*)

- * V primeru, da imamo opravka s produktom, se izvede analiza vrzeli, analiza integracije izvirnega sistema z drugimi sistemi in analiza strategij za odklop starega sistema iz uporabe sočasno z aktivnostjo analize razvoja produktne programske opreme. Tako se lahko zmanjša vložek, ker se analiza naredi samo enkrat.
- o Dokončaj seznam migracijskih zahtev (angl. *Finalize Migration Requirements*)
 - * Glede na analizo se prilagodi zahteve pridobljene v fazi inicializacije.
- o Določi obseg migracije (angl. *Determine To Be Migrated Data*)
 - * Podrobno se opredeli, kateri podatki se bodo migrirali. Posebno pozornost je potrebno posvetiti vprašanju, ali se migrirajo tudi zgodovinski podatki. Mogoče možnosti so:
 - ◇ Zgodovinske podatke se migrira.
 - ◇ Zgodovinske podatke se arhivira.
 - ◇ Zgodovinske podatke se zavrže.
 - Poleg tega je potrebno določiti, kakšna je še sprejemljiva stopnja redundance migriranih podatkov, kar je odvisno od poslovnih pravil ciljnega sistema. Za aktivnost so zadolženi vodje poslovnih področij, poslovni analitiki in skrbniki podatkovnih baz.
- o Določi strategijo testiranja migracije (angl. *Define Testing Strategy*)
 - * Podrobno se popiše postopek testiranja rezultata migracije.
- o Določi strategijo izvedbe migracije (angl. *Define Deployment Strategy*)
 - * Podrobno se popiše postopek za izvedbo migracije. Strategija je odvisna od obsega podatkov, kompleksnosti sistema in tehničnih omejitev ter določenega časovnega okvira, v katerem naj bi se izvedla migracija.
- o Razširi strategijo za testiranje produkta (angl. *Extend Product Testing Strategy*)

- * V primeru, da imamo opravka s produktom, se izvede testiranje rezultata migracije v okviru testiranja produkta.
 - Razširi strategijo za vpeljavo produkta (angl. *Extend Product Deployment Strategy*)
 - * V primeru, da imamo opravka s produktom, izvedemo migracijo v okviru uvedbe produkta v produkcijsko okolje.
 - Pridobi odobritev s strani vodij poslovnih področij (angl. *Get Business Approval*)
 - * S to aktivnostjo zagotovimo, da so vodje poslovnih področij vključeni v fazi analize.
- Razvoj (angl. *Development*)
 - Načrtuj (angl. *Design*)
 - * Določimo, kako bo programska rešitev za migracijo podatkov izgledala. Aktivnost je enaka aktivnosti, ki se jo izvaja za tipično načrtovanje razvoja programske opreme.
 - Zgradi programsko opremo za migracijo podatkov (angl. *Build The Migration Software*)
 - * Aktivnost je enaka tipičnemu razvoju programske opreme.
 - Testiraj (angl. *Test*)
 - * Razvijalci pretestirajo programsko opremo za migracijo podatkov.
 - Pridobi odobritev s strani vodij poslovnih področij (angl. *Get Business Approval*)
 - * S to aktivnostjo zagotovimo, da so vodje poslovnih področij vključeni v fazi razvoja.
- Testiranje (angl. *Testing*)
 - Testiraj izvajanje migracije podatkov (angl. *Test Migration Run*)

- * V tej aktivnosti preverimo, da se migracija procesno izvede brez napak.
- Testiraj celovitost podatkov (angl. *Test Completeness*)
 - * Vsi zahtevani podatki so bili v celoti prenešeni.
- Testiraj konsistentnost podatkov (angl. *Test Consistency*)
 - * Vse povezave med podatki po migraciji so ohranjene, prav tako podatkovni tipi migriranih podatkov.
- Testiraj izgled podatkov (angl. *Test Appearance*)
 - * Podatki so pravilno in v skladu s pričakovanji vidni v uporabniškem vmesniku novega sistema.
- Testiraj procesiranje podatkov (angl. *Test Processability*)
 - * Migrirani podatki ne smejo ogroziti delovanja kateragakoli procesa, ki se izvaja na novem sistemu.
- Testiraj integracije (angl. *Test Integration*)
 - * Vsi integrirani sistemi, ki bodo komunicirali z novim sistemom, morajo imeti možnost dostopati do migriranih podatkov. Prav tako podatki ne smejo ogroziti delovanja kateregakoli procesa integriranih sistemov.
- Pridobi odobritev s strani vodij poslovnih področij (angl. *Get Business Approval*)
 - * S to aktivnostjo zagotovimo, da so vodje poslovnih področij vključeni v fazi testiranja.
- Izvedba (angl. *Deployment*)
 - Izdelaj izvedbeni projektni plan (angl. *Develop Deployment Project Plan*)

- * Pripravi se podroben plan, v katerem so popisani koraki, po katerih se bo izvajala migracija podatkov. Ker je migracija ponavadi vezana tudi na vpeljavo prenovljenega sistema, moramo pri vrstnem redu korakov upoštevati tudi korake pri vpeljavi novega sistema v produkcijsko okolje.
- o Izvedi zadnjo testno migracije podatkov (angl. *Perform Rehearsal*)
 - * Preden izvedemo migracijo na produkcijskem okolju moramo izvesti testno migracijo na izvornih podatkih ali vsaj na okolju, ki je čim bolj podobno produkcijskemu. Večja kot je podobnost, večja je verjetnost, da bomo zaznali morebitne probleme preden migracijo izvedemo v produkcijskem okolju.
- o Pridobi soglasje lastnikov podatkov za migracijo v produkcijsko okolje (angl. *Get Data Owner Go Ahead*)
 - * Če so lastniki podatkov zadovoljni z rezultatom migracije (validacija) iz prejšnje aktivnosti, lahko z uradno odobritvijo nadaljujemo z izvedbo migracije v produkcijskem okolju.
- o Izvedi migracijo (angl. *Execute The Migration*)
 - * Izvedba migracije produkcijskih podatkov v produkcijskem okolju.
- o Vrni sistem na prejšnje stanje (angl. *Fallback*)
 - * Če se migracija ne izvede pravilno, moramo imeti pripravljen postopek, kako vrniti izvorna podatkovna skladišča in izvorni sistem nazaj v uporabo v čim krajšem možnem času.
- o Preklopi na nov sistem (angl. *Cut-Over To Target*)
 - * Če se migracija izvede uspešno, uporabimo strategijo za odklop starega sistema in podatkovnih skladišč iz uporabe, ki smo jo določili v fazi analize
- o Nadzoruj delovanje sistema po izvedbi migracije (angl. *Monitor Post-Deployment*)

- * Po izvedbi migracije je potrebno spremljati delovanje sistema in popraviti napake v primeru, da ugotovimo, da podatki manjkajo ali so napačno zmigrirani.

Idu je metodo preizkusil tudi v praksi ter opravil intervjuje z ljudmi, ki so preverili ustreznost metode pri svojem delu. Izkazalo se je, da procesi, ki jih je Idu izluščil iz obstoječe literature, zelo dobro pokrijejo aktivnosti na projektu migracije podatkov. Zato smo se odločili, da tudi mi poskusimo slediti korakom, ki jih predlaga Idu, in na koncu ocenimo, kako celovito lahko z njimi pokrijemo naše potrebe in potrebe naročnika. Ugotovitve so predstavljene v poglavju Metodologija migracije podatkov.

Poglavje 4

Metodologija migracije podatkov

V okviru vpeljave novega informacijskega sistema kot produkta je bila izvedena tudi migracija pri treh različnih naročnikih v razdobju treh let. Obseg migriranih podatkov se je razlikoval od naročnika do naročnika, prav tako preslikava podatkov, saj je bil podatkovni model starih sistemov različen. Strategijo migracije podatkov je bilo potrebno prilagoditi vsakemu naročniku. Za osnovo smo vzeli kombinacijo metod Migracija podatkov za podatkovno intenzivno produktno programsko opremo, ki jo je predstavil Idu, in Procesni model, avtorjev Matthes, Schultz in Haller (2011), ter prilagodili aktivnosti potrebam našega projekta. Tako je proces migracije podatkov razdeljen na pet faz kot kaže Slika 4.1.

Aktivnosti, ki so bile opravljene v okviru krovnega projekta vpeljave novega sistema, npr. Javni razpis in ponudba, ne bodo obravnavane posebej, razen v primeru, ko bo to pomembno tudi za proces migracije podatkov.

Izvoz izvornih testnih podatkov in Izvoz izvornih podatkov je v našem primeru izvajal ali naročnik ali lastnik podatkov integriranega sistema, vendar smo koraka vseeno umestili v faze migracije podatkov.

Seznam dokumentov, ki so značilni za posamezno fazo, je prikazan v tabeli pri vsaki fazi.

DNEVI	22.feb	29.feb	7.mar	14.mar	21.mar	28.mar	4.apr	11.apr	18.apr	25.apr	2.maj	9.maj	16.maj	23.maj	
											Uvedba novega sistema				
MIGRACIJA PODATKOV															
Visokoravenski pogled na projektni plan migracije podatkov	Inicilizacija														
		Analiza													
				Razvoj											
				Testiranje											
								Izvedba							
TESTIRANJE															
Testna skupina izvajalca migracije						Testiranje (Mar 23th)	Testiranje (Mar 31st)	Testiranje (Apr 5th)	Testiranje (Apr 11th)						
OKOLJA															
		LOCAL	LOCAL	LOCAL	TEST	TEST	TEST	STAGE	STAGE	STAGE	PROD	PROD	PROD	PROD	

Slika 4.1: Visokoravenski pogled na projektni plan migracije podatkov.

4.1 Faza 1: Inicijalizacija

Seznam korakov faze Inicijalizacija je viden na Sliki 4.2.

DNEVI	22.feb	29.feb	7.mar
MIGRACIJA			
Inicijalizacija	Določitev vseh deležnikov podatkov	Predhodna analiza (Analiza stanja in analiza vpliva na poslovanje)	Pridobivanje zahtev za migracijo
		Določitev strategije migracije podatkov	Izdelava projektnega plana ter ocena potrebnega časa in stroškov
			Vzpostavitev okolja

Slika 4.2: Faza Inicijalizacija.

4.1.1 Določitev vseh deležnikov podatkov

Dokumentiran je bil seznam ljudi, ki sodelujejo pri migraciji podatkov ali pa migracija na njih neposredno ali posredno vpliva. Migracija se ponavadi ne izvaja samostojno, pač pa je le ena od aktivnosti, ki dopolnjuje zamenjavo, nadgradnjo ali združitev več sistemov. Večinoma je tesno povezana in odvisna od ostalih aktivnosti, zato morajo biti vsi, ki so kakorkoli vpleteni pri migraciji, obveščeni o stanju ostalih aktivnosti (zakasnitve pri postavitvi

novega testnega sistema, kjer naj bi se izvajala testna migracija in verifikacija novega sistema, pomembne, novo dodane funkcionalnosti poslovnega sistema in s tem sprememba podatkovne sheme, nadgradnja verzije sistema za upravljanje s podatkovnimi bazami, itd.). Pomembno je, da je seznam čim bolj popoln, zato moramo dokumentacijo ažurirati v primeru, da se seznam deležnikov spremeni. To velja predvsem, ko imamo opravka z daljšim projektom migracije podatkov.

Nivo in pogostost obveščanja je odvisna od kritičnosti problema in se ju lahko sproti prilagaja. V praksi se je izkazalo, da so tedenski sestanki primerni, kadar problemov ni veliko oziroma niso kritični. V nasprotnem primeru je potrebno razmisliti o dnevni sestankih, da se lahko težave rešujejo sproti in se ne nabere preveč do naslednjega tedenskega sestanka.

Način obveščanja je odvisen od deležnikov. Na nižjem nivoju, kjer se problemi rešujejo, je najlažje sklicati sestanek, medtem ko se ostalim pošlje povzetek problematike in dogovorov v pisni obliki. Stanje se redno predstavi tudi na projektnih sestankih. Take sestanke je dobro imeti ob večjih mejnikih, v primeru agilne metode ob koncu vsakega sprinta. Povzetke s sestankov je potrebno v neki obliki shraniti na mesto, ki je dostopno vsem sodelujočim na projektu, da se preveri dogovore, v kolikor pride do dilem ali celo nesoglasij.

Pri vseh petih migracijah, kjer smo sodelovali, se je zatikalo pri komunikaciji. Opazili smo, da je bilo obveščanje boljše, če so bili deležniki bolj povezani med sabo, torej da komunikacija med njimi ni bila le striktno formalna.

Pri prvem naročniku smo imeli največ problemov z lastniki podatkov integriranih sistemov, saj je bila komunikacija omejena izrecno samo na statusne sestanke, zato izvoženi podatki, ki smo jih zahtevali od njih, velikokrat niso bili v skladu z dogovorom, posledica pa je bila napaka v delovanju novega sistema po njegovi uvedbi.

Pri drugem naročniku smo največ časa izgubili, ker večinoma nismo imeli dostopa do podatkov, pač pa je rezultate na podatkovni bazi in testiranje same migracije podatkov izvajal skrbnik podatkovnih baz na strani naročnika, kar

je zavleklo aktivnosti.

Največja težava pri tretjem naročniku pa je bila veliko število deležnikov, saj je bil projekt kompleksen, in je bilo zato obveščanje vseh vpletenih oteženo. Dodatne nevšečnosti smo imeli z oddelkom za varnostno politiko, ki je potreboval veliko časa za odobritev dostopa do produkcijskih podatkov oziroma dostopa do različnih okolij.

4.1.2 Predhodna analiza (Analiza stanja in analiza vpliva na poslovanje)

Preden se migracija lahko začne, je potrebno raziskati, kakšno je trenutno stanje sistema, kakšno je željeno stanje in obseg vrzeli med obema sistemoma. Treba je paziti, da sta stanji čimbolj natančno opisani, saj je od tega odvisno, ali je migracija (delna ali v celoti) sploh smiselna. Lahko se zgodi, da je razlika med sistemoma prevelika, da bi lahko migracijo opravili v časovnem in stroškovnem okviru, ki nam je dan na razpolago, ali pa bi bila lahko škoda zaradi nedelovanja sistema med migracijo prevelika.

4.1.3 Pridobivanje zahtev za migracijo

V sodelovanju z naročnikom in poslovnimi analitiki je potrebno določiti seznam podatkov, ki bi jih radi prenesli na nov sistem. Obstaja velika verjetnost, še posebej če je migracija podatkov obsežna, da vse zahteve in omejitve še ne bodo v celoti pokrite v tej fazi ali pa se bodo zahteve še spreminjale. Vseeno pa je dobro, da se definira čim več, da se lahko nato nadaljuje s preostalimi aktivnostmi.

V našem primeru, kjer smo vpeljali nov produkt, je bilo veliko odvisno od stanja na projektu razvoja novega sistema. Ker se je aplikacija še razvijala, s tem pa se je tudi podatkovni model še spreminjal, je bilo težko določiti, katere od podatkov bo moč migrirati in v kakšni obliki. To je bilo predvsem vidno pri prvem naročniku. Pri naslednjih dveh se je sistem manj spreminjal in je bil zato vložen trud manjši.

Potrebno je določiti:

- Poslovne zahteve,
 - poslovne domene, ki naj se migrirajo,
 - pomen poslovnih objektov in relacije med njimi,
 - količina, tip in prioriteta migriranih podatkov,
 - časovni okvir, v katerem se sistem ustavi (če je potrebno) in se izvede migracija,
 - seznam uporabnikov novega sistema, ki morajo biti obveščeni o novih funkcionalnostih sistema in morebitnih izpadih, ter način obveščanja in šolanja uporabnikov,
 - dovoljen čas izpada sistema med migracijo podatkov in ob morebitnih problemih po njej ter odškodnina v primeru prekoračitve tega časa,
 - kriteriji za prevzem novega sistema in predajo migriranih podatkov lastnikom podatkov.
- Tehnične zahteve (strojna in programska oprema, ki se bo uporabila pri migraciji),
 - S pomočjo podatka o obsegu migracije in ocene, koliko novih podatkov se bo zgeneriralo v nekem časovnem obdobju, se opredelijo zahteve za strojno opremo (procesorska moč, velikost in tip pomnilnika, velikost trdih diskov, pretočnost podatkov skozi omrežje, potrebna oprema za varnostne kopije, itd.).

Pri programski opremi, ki je zahtevana ali zaželenja, ne smemo pozabiti na pravočasno pridobitev licenc, če je to potrebno, sicer bi se lahko projekt zavlekel. Upoštevati je potrebno tudi programsko opremo, ki bo potrebna za ugotavljanje problemov (npr. programska oprema za spremljanje performančnih zmogljivosti sistema) po vpeljavi novega sistema.

Rezultat tega koraka je dokument s popisom strojne in programske opreme, opisom namembnosti, lastništva, uporabnikov, ki imajo dostop do opreme, in njihovih pravic ter stanjem licenc.

- Operativne zahteve (zahteve glede varnosti in skladnosti s standardi).
 - seznam udeležencev na projektu, njihova vloga na projektu ter pravice in dostopi, ki jih potrebujejo za svoje delo,
 - način, kako bodo izvoženi podatki predani skupini zadolženi za migracijo podatkov,
 - postopek, kako bo testna skupina obveščala udeležence na projektu migracije podatkov in po potrebi tudi udeležence krovnega projekta razvoja novega sistema, če bi našla problematične podatke v novem sistemu,
 - zadolžitve pri zagotavljanju delovanja sistema ter postopek obveščanja o izpadu sistema,
 - način izdelave varnostnih kopij podatkov,
 - urnik izvajanja nadgrajenij operacijskega sistema in ostale programske opreme.

Popis stanja, seznam željenih in potrebnih sprememb ter najdenih odstopanj bodo potrebni v kasnejši aktivnosti, Izdelava projektnega plana ter ocena potrebnega časa in stroškov, kjer se na osnovi zahtev pripravi projektni plan.

4.1.4 Določitev strategije migracije podatkov

Omenili smo že, da je strategija migracije odvisna od projekta, predvsem od zahtev in omejitev. Ker migracija podatkov ni le prenos podatkov z enega na drug sistem, temveč je pred prenosom potrebno podatke preslikati v pravo obliko in določiti vrednosti, če jih ni (za kar je potrebno dobro poznati oba poslovna sistema in način shranjevanja podatkov), ter se čimprej odločiti za

strategijo migracije podatkov. Robert Levine [9] je sicer opisal štiri možne strategije migracije, vendar smo združili dva načina v enega (migracija po fazah in migracija po logičnih celotah) [10]:

- Migracija v enem koraku ("big bang").

Vsi podatki se migrirajo naenkrat v časovnem oknu, ki ga določijo vodje poslovnih področij. Takrat je poslovni sistem ugasnjen, zato ni potrebno skrbeti za sinhronizacijo podatkov. Ta način je preprostejši, vendar je reševanje težav težje, saj operiramo nad vsemi podatki. Tudi vrnitev sistema v stanje pred migracijo je težje zaradi količine podatkov.

Prednosti:

- Preprostejši za razvoj,
- nižji stroški,
- migracija podatkov se izvede samo enkrat,
- uporabniki uporabljajo le nov sistem, vnašanje v več sistemov kot pri paralelnem delovanju ni potrebno.

Slabosti:

- Poslovni sistem mora biti ugasnjen,
- veliko tveganje (vrnitev sistema v prejšnje stanje je težje zaradi količine podatkov),
- majhne napake ali podrobnosti se lahko spregledajo zaradi količine podatkov,
- pazljivo je potrebno določiti reprezentativni vzorec podatkov, ki bodo verificirani po migraciji,
- napaka na nekem delu sistema lahko kritično vpliva na delovanje drugih delov sistema,
- uporabniki morajo znati uporabljati cel sistem takoj, ko se ga da v uporabo.

- Migracija po fazah (najprej pomembni podatki, nato manj pomembni podatki ali zgodovinski podatki, migracija po logičnih celotah - po integriranih sistemih, operacijskih sistemih, bazah, poslovnih funkcijah, itd.)

Imenujemo jo tudi iterativna migracija, saj postopoma dodajamo podatke v sistem. S tem načinom zmanjšamo tveganje, ker je podatkov manj, hkrati pa migriramo le omejen del podatkov poslovnega sistema, kar pomeni, da preostali del sistema ne bo prizadet. Čeprav zmanjšamo kompleksnost pri preslikavi podatkov in je sama migracija podatkov v produkcijsko okolje hitrejša, pa se lahko zaplete pri spregledanih odvisnostih, ki veljajo med temi podatki in podatki drugih delov sistema.

Prednosti:

- Manjše tveganje, ker je manj podatkov,
- ko najdemo manjše napake, se jih da odpraviti sproti,
- z iterativnim izvajanjem migracij in pridobivanjem znanja lahko izpolnimo proces migracije in validacije migriranih podatkov,
- učenje uporabe sistema je lahko postopno.

Slabosti:

- Potrebno je dobro poznavanje poslovanja in podatkov, podatkovnega modela ter odvisnosti med podatki izvirnega sistema, da lahko določimo, katere podatke je mogoče med sabo povezati in zbrati v skupino ter jih migrirati skupaj,
- čas za razvoj orodja za migracij podatkov je lahko daljši,
- ker migracija podatkov ni izvedena enkratno, se lahko zgodi, da se dogovori o določenih preslikavah pozabijo,
- lahko se zgodi, da se z vsako nadaljno fazo interes deležnikov migracije podatkov manjša in kvaliteta migriranih podatkov zelo pade,

- ko so podatki odvisni eden od drugega, vendar jih ne moremo migrirati istočasno, se lahko zgodi, da nekateri podatki pridejo v sistem šele, ko so tudi odvisni podatki migrirani,
- ker se podatki uporabljajo tako v novem kot starem sistemu, je potrebno aplikacijo za dostop do njih prilagoditi tako, da dostopa do obeh sistemov, kar poveča kompleksnost sistema.
- Migracija s paralelnim delovanjem starega in novega sistema (kot potrditeljev koncepta ali zmanjšanje tveganja)

Vse podatke prenesemo naenkrat, vendar oba sistema delujeta paralelno, s čimer dopustimo daljše testiranje novega sistema ter naročniku omogočimo, da tudi po migraciji lahko zazna napake pri določevanju preslikave ali procesu migracije. Uporabniki morajo vnašati spremembe v oba sistema ali pa razvijemo rešitev, ki sinhronizira podatke obeh sistemov (enosmerno ali v obeh smereh).

Prednosti:

- Manjše tveganje, ker imamo oba sistema delujoča in izvorni podatki so vedno na voljo,
- vodje poslovnih področij lahko validirajo migrirane podatke šele, ko so zares prepričani, da sistem deluje v skladu s pričakovanji.

Slabosti:

- Večji strošek, ker moramo vzdrževati dva sistema,
- uporabniki imajo več dela, ker vnašajo v dva sistema, ali pa je potreben dodaten trud za razvoj rešitve za sinhronizacijo sistemov.

V praksi se največkrat izkaže, da se strategije med sabo kombinirajo, saj se lahko tako migracija podatkov najlažje prilagodi naravi poslovnega sistema in količini ter tipu podatkov.

Odločili smo se, da bomo orodje za migracijo napisali sami, sestavljeno pa je iz poizvedb SQL. Orodje lahko s prilagoditvijo uporabimo tudi pri naslednjih migracijah, saj ima produkt enako podatkovno shemo za vse naročnike oziroma se le počasi spreminja z nadgradnjami. Z orodjem je mogoča tudi inkrementalna migracija (angl. *Delta migration*), kjer se migrirajo le tisti podatki z izvirnega sistema, ki so se dodali ali so se spremenili od zadnje migracije. Narejeno je modularno, tako da je moč na enostaven način omogočiti ali onemogočiti migracijo podatkov določene poslovne funkcije. Orodje vsebuje tudi poizvedbe SQL za hitro verifikacijo celovitosti in pravilnosti podatkov.

Dokler se orodje za migracijo še razvija ali prilagaja novim pravilom preslikave ali preoblikovanja, lahko testno migracijo izvajamo na podatkih, ki so naključno generirani. Najprej se izvede migracija podatkov v celoti, nato sledi ena ali dve inkrementalni migraciji. Ta cikel ponavljamo, dokler nismo zadovoljni z rezultatom migracije oziroma, v praksi se izkaže, dokler nam časovne omejitve to omogočajo.

Za verifikacijo vsakega paketa izvoženih podatkov so bili zadolženi testerji na naročnikovi strani ali na strani integracijskega sistema. Verifikacija migriranih podatkov pa je bila opravljena z naše strani. Tako je vsak prevzel odgovornost za celovitost in pravilnost podatkov na svojem delu.

Sledi testna izvedba migracije v celoti na okolju, ki je čim bolj podobno produkcijskemu okolju (angl. *Stage environment*), če je izvedljivo, na produkcijskih podatkih, ki niso anonimizirani. Tako preverimo, da so podatki celoviti in konsistentni ter da je bila izvedba migracije na produkcijskih podatkih uspešna. Zato lahko tak rezultat pričakujemo tudi v produkcijskem okolju. Edino, kar nam lahko povzroči težave, so že obstoječi podatki v ciljnem sistemu, če ti obstajajo. V primeru, ko je to dovoljeno, je najlažje stestirati migracijo na kopiji produkcijske podatkovne baze (s produkcijskimi podatki, ki niso anonimizirani).

Ko vodje poslovnih področij potrdijo, da so zadovoljni z rezultatom migracije, smo pripravljeni na migracijo v produkcijsko okolje.

Pri prvem naročniku smo izvedli migracijo dvakrat, in sicer po logičnih celo-

tah, podatkov pa je bilo malo. Med izvajanjem migracije sistem ni deloval. Za drugo migracijo je bilo potrebno skleniti dogovor, kako se bo reševalo primere, kjer so podatki v sistemu in podatki, ki se migrirajo, podvojeni. Časovno okno za migracijo je bilo dovolj veliko, da smo izvedli migracijo podatkov brez posebnih tehničnih optimizacij. Zgodovinskih podatkov nismo migrirali.

Drugi naročnik je zahteval večji obseg migracije in časovno okno za migracijo je bilo opazno manjše kot pri prvem naročniku, zato je bilo potrebno orodje optimizirati. Migracija je potekala v enem koraku. Izvajali je nismo mi, temveč naročnikov skrbnik podatkovnih baz, kar je zapletlo fazo izvedbe. Tudi tu zgodovinski podatki niso bili zmigrirani.

Z največjim obsegom podatkov smo imeli opravka pri tretjem naročniku, zato smo se odločili, da bomo izvedli migracijo hibridno med strategijama Paralelno delovanje in Dostava po fazah. Najprej smo zmigrirali pomembne podatke enega integriranega sistema. Z migracijo podatkov v celoti smo začeli en teden pred rokom in izvedli dodatni dve inkrementalni migraciji do roka. V časovnem oknu, ki smo ga imeli za migracijo, ko sistem ni deloval, smo izvedli le inkrementalno migracijo na majhnem številu podatkov, ostalo je bilo narejeno že prej. Po uspešni migraciji smo nov sistem dali v uporabo, ki pa je deloval paralelno s starim sistemom, zato je bilo potrebno produkt nadgraditi s funkcionalnostjo za sinhronizacijo podatkov med starim in novim sistemom. Sledila je migracija drugega integriranega sistema, kjer pa je bilo podatkov več. Prav tako smo začeli z migracijo v celoti na sistem teden pred rokom, a sistema nismo ugasnili. Zopet sta sledili dve inkrementalni migraciji. Pri zadnji inkrementalni migraciji podatkov je bil sistem ugasnjen, saj je bil tudi nadgrajen. Enak postopek je bil izveden pri integraciji ostalih integriranih sistemov. Migraciji zgodovinskih podatkov smo se izognili pri tem projektu, saj bi samo analiza podatkov izvirnega sistema trajala predolgo in časovnim omejitvam ne bi mogli zadostiti.

4.1.5 Izdelava projektnega plana ter ocena potrebnega časa in stroškov

Migracija podatkov ponavadi ni edina aktivnost, ki se izvaja, ko se uvaja nov, prenovljen sistem ali pa se združuje več sistemov. Vodi se sicer kot samostojen projekt, vendar je odvisna od drugih načrtovanih aktivnosti uvajanja novega poslovnega sistema.

Za uspešno izvedbo migracije je potrebno dobro poznati tako poslovni sistem, ki je izvor podatkov, kot tudi nov poslovni sistem, kamor se bodo podatki prenesli, zato mora skupina, ki je zadolžena za migracijo, tesno sodelovati z ostalimi projektnimi člani. Poskrbeti moramo, da so vsi predpogoji za migracijo podatkov pravočasno izvedeni (da so testna okolja, na katerih se bodo izvajale testne migracije na voljo, da je skupina testerjev zadolžena za verifikacijo na voljo, itd.).

Z dobro pripravljenim planom lahko vnaprej predvidimo nekatere probleme, popišemo njihov postopek odprave in tako zmanjšamo možnosti za kasnejše težave.

Pri izdelavi projektnega plana je potrebno določiti:

- Vloge in naloge pri migraciji podatkov:
 - Vodja projekta: bedi nad projektom in usklajuje udeležene pri migraciji, poroča o trenutnem stanju in o morebitnih problemih.
 - Vodje poslovnih področij:odobrijo nadaljevanje po vsaki fazi projekta migracije podatkov ter validirajo podatke in prevzamejo sistem.
 - Poslovni analitik na strani vira podatkov: poznavanje starega sistema.
 - Poslovni analitik na strani izvajalca migracije: poznavanje funkcionalnosti novega sistema, pomoč pri sporazumevanju z naročnikom.
 - Razvijalec podatkovnih baz, na strani vira podatkov: zadolžen za razvoj orodja za izvoz podatkov iz starega sistema.
 - Razvijalec podatkovnih baz: zadolžen za razvoj orodja za vnos podatkov v nov sistem.

- Razvijalec starega poslovnega sistema: poznavanje podrobnosti obnašanja stare aplikacije.
- Razvijalec novega poslovnega sistema: poznavanje podrobnosti funkcionalnosti novega sistema in sledenje novim funkcionalnostim in spremembam podatkovne sheme, če je novi sistem še v fazi razvoja.
- Tester na strani izvora podatkov: Poznavanje podatkov in verifikacija in validacija izvoženih podatkov.
- Tester na strani izvajalca migracije: poznavanje podrobnosti funkcionalnosti novega sistema in verifikacija podatkov in delovanja sistema po migraciji.
- Skupina za upravljanje s konfiguracijami: nameščanje novih verzij.
- Skrbnik podatkovnih baz: omogoča pristope do baze v okviru potreb in varnostnih praks, pomaga izboljšati zmogljivost postopka migracije, javlja ter pomaga reševati morebitne težave na podatkovnem strežniku in izvede migracijo podatkov.
- Sistemski administrator: ponastavlja pravice in pristope na sistemskem nivoju za potrebe migracije, skrbi za nameščanje posodobitev sistema in njihovo pravilno delovanje.
- Podrobni plan migracije, potek dela (sosledje akcij in uporabe virov ter dodelitev vlog):
 - Priprava podrobnega plana testnih migracij.
 - Priprava podrobnega plana produkcijske migracije.
- Plan za testiranje podatkov in aplikacije po migraciji (verifikacija).
- Varnostno politiko na nivoju projekta:
 - Določitev režima uporabe različnih okolij (razvojno, testno, stage, produkcijsko).
 - Določitev režima dostopa do realnih podatkov (anonimizacija občutljivih podatkov, omejitev dostopa).

Pri vseh treh naročnikih se je izkazalo, da je pri planiranju potrebno upoštevati, da je migracija podatkov ponavadi podcenjena in zato so časovni roki za izdelke precej kratki. Če je možno, se aktivnosti izvaja sočasno, saj tako pridobimo čas. Upoštevati moramo tudi, da je projektni plan migracije del plana uvedbe novega sistema, kjer pa se lahko katerakoli faza zavleče, kar bo seveda imelo vpliv tudi na projekt migracije. Zato, če je možno, pri planiranju migracije vnesemo tudi dodaten čas, ki nam bo služil za zamik aktivnosti v primeru zamujanja krovnega projekta uvedbe novega sistema.

Ko je planiranje končano, je potrebno ponovno oceniti [15]:

- Ali so vsi časovni roki in izdelki jasno opredeljeni?
- Ali so vsi stroški še vedno pokriti?
- Ali smo vključili vse morebitne udeležence na projektu v plan?
- Ali smo opredelili plan komunikacije in ali smo vključili vanj vse deležnike, upravo in, če je potrebno, širšo organizacijo?
- Ali bo na voljo dovolj ljudi s pravimi kompetencami tekom projekta? Bolj natančno, ali imamo zadosti:
 - Poslovnih analitikov;
 - Strokovnjakov za migracijo podatkov;
 - Skrbnikov podatkovnih baz;
 - Sistemskih administratorjev;
 - Testerjev.

4.1.6 Vzpostavitev okolja

Čim prej moramo doseči dogovor, kje in kdaj se bo izvajala migracija podatkov, da se ne prekriva z drugimi aktivnostmi, ki se ne morejo izvajati sočasno z migracijo. Poleg tega je potrebno paziti, da migrirani podatki ostanejo prisotni na sistemu toliko časa, dokler se testiranje ne izvede do konca, šele po tem jih lahko umaknemo.

Pri drugem naročniku dogovorjeno okolje ni bilo na voljo za migracijo podatkov zaradi zamujanja krovnega projekta uvedbe novega sistema. V drugem primeru pa smo morali migrirane podatke zbrisati s testnega sistema zaradi varnostne politike, še preden je bilo izvedeno testiranje v celoti. Tako je bilo izvedeno manj testnih migracij in testiranj migriranih podatkov kot je bilo predvideno, kar je povečalo tveganje za neuspeh projekta migracije podatkov. Zaradi integriranih sistemov pri tretjem naročniku testiranje procesiranja podatkov in integracijski test ni bil mogoč, saj so bili integrirani sistemi povezani le na produkcijskem okolju. Posledično je bilo tudi tu tveganje za neuspeh večje.

4.1.7 Odobritev s strani vodij poslovnih področij

Čeprav je to pomemben korak, pa se velikokrat zgodi, da se ga enostavno izpusti, ker lastniki podatkov niso na voljo ali pa nimajo interesa. Kljub temu je potrebno vztrajati, da se odgovorni strinjajo z vsemi aktivnostmi te faze in odobrijo nadaljevanje z naslednjo fazo migracije podatkov.

Na žalost so se v našem primeru pri tretjem naročniku zahteve spremenile pravzaprav na dan migracije, zato tudi ni bilo pričakovati, da bomo deležni odobritve, in smo aktivnost preskočili.

Seznam dokumentov, ki so značilni za to fazo prikazuje Tabela 4.1.

Korak	Dokument
Določitev vseh deležnikov podatkov	Dokument s seznamom vseh deležnikov podatkov
Predhodna analiza	Poročilo o predhodni analizi stanja in vplivu migracije podatkov na poslovanje
Pridobivanje zahtev za migracijo	Začetna oblika dokumenta s seznamom zahtev (poslovnih, tehničnih in operativnih) za migracijo podatkov ter seznam omejitev
Določitev strategije migracije podatkov	Dokument z opisom strategije migracije podatkov
Izdelava projektnega plana ter ocena potrebnega časa in stroškov	Projektni plan migracije ter dokument z oceno potrebnega časa in stroškov za izvedbo projektnega plana migracije podatkov ter zahtevami vseh udeležencev na projektu
Vzpostavitev okolja	Dokument s seznamom in opisom vseh okolij ter matrika dostopov udeležencev projekta do teh okolij
Odobritev s strani vodij poslovnih področij	Pisna odobritev za nadaljevanje projekta migracije podatkov

Tabela 4.1: Dokumenti faze Inicializacije

Vloge udeležencev na projektu pri fazi Inicializacija so razvidne iz Tabele 4.2 in Tabele 4.3.

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemeski administrator
Določitev vseh * deležnikov podat- kov													
Predhodna analiza (Analiza stanja in analiza vpliva na poslovanje)	*	*	*	*	*	*	*	*				*	
Pridobivanje zah- tev za migracijo	*	*	*									*	*

Tabela 4.2: Koraki v fazi Inicializacija in vloge udeležencev na projektu (1. del)

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemske administrator
Določitev strategije migracije podatkov	*	*	*	*								*	*
Izdelava projektnega plana ter ocena potrebnega časa in stroškov	*	*	*	*	*	*	*	*	*	*	*	*	*
Vzpostavitev okolja										*	*	*	*
Odobritev s strani vodij poslovnih področij		*											

Tabela 4.3: Koraki v fazi Inicializacija in vloge udeležencev na projektu (2. del)

4.2 Faza 2: Analiza

Seznam korakov faze Analiza je viden na Sliki 4.3.

DNEVI									
29.feb			7.mar			14.mar			
MIGRACIJA									
Analiza	Analiza izvornega sistema	Analiza ciljnega sistema	Analiza vrzeli	Analiza integracije z drugimi sistemi	Določitev strategije za odklop starega sistema iz uporabe	Dokončna določitev zahtev	Določitev obsega migracije	Določitev strategije izvedbe migracije	Določitev strategije testiranja migracije

Slika 4.3: Faza Analiza.

4.2.1 Analiza izvornega sistema

Ko pridobimo osnovne zahteve za migracijo, moramo raziskati, kje se podatki v izvornem sistemu nahajajo. S pomočjo poslovnih analitikov in lastnikov podatkov moramo določiti pomen različnih podatkov. To je idealni čas za čiščenje podatkov, ker bomo tako imeli veliko manj različnih vrednosti, za katere moramo zagotoviti, da jih pravila preslikave zaobjamejo.

Ta proces zna biti zelo kompleksen in dolgotrajen, ker izvorna podatkovna skladišča, iz katerih izvažamo podatke za migracijo, niso ali niso dobro dokumentirana. V najslabšem primeru je potrebno poseči celo po povratnem inženirstvu, da se ugotovi pomen podatkov in se identificira poslovne procese, ki uporabljajo te podatke.

Zanimivo je, da so se napake zaradi izvoza napačnih podatkov pokazale šele kasneje, ko je bil nov sistem že dan ali celo več v uporabi, zato smo morali migrirane podatke popravljati na produkcijskem okolju.

Čeprav smo na novejših projektih upoštevali to tveganje, vpeljali smo celo dodatno preverjanje izvoženih podatkov, se je izkazalo, da napake nismo uspeli zajeti. Šele po nekaj mesecih je naročnik opazil, da nekaj podatkov manjka. S poslovnega vidika podatki sicer niso bili kritični, sicer bi jih opazili že prej, vendar pa je bila vseeno storjena napaka pri določanju, kateri podatki morajo biti izvoženi in migrirani. Da bi popravili škodo, smo izvedli še dve manjši migraciji podatkov.

4.2.2 Analiza ciljnega sistema

Ko vemo, kaj je potrebno zmigrirati, se analizira ciljni sistem. Predvsem nas zanima, kam bomo shranili podatke ter v kakšni obliki.

V našem primeru, ko smo vpeljevali sistem kot produkt, smo največ časa za analizo ciljnega sistema porabili pri prvem projektu migracije. Ko so bile relacije med podatki enkrat določene in ko smo vedeli, kateri podatki so obvezni za pravilno delovanje sistema, je bil razvoj orodja za migracijo podatkov oziroma njegovo prilaganje za nov projekt hitro.

4.2.3 Analiza vrzeli

Pri naših projektih je ta del trajal najdlje. Ugotoviti je potrebno, ali izvoženi podatki ustrezajo formatu, v katerem naj bi bili shranjeni na ciljnem sistemu, ali bosta potrebni preslikava in preoblikovanje podatkov.

Nekateri podatki se med sistemoma ne razlikujejo, predvsem šifranti, ki so narejeni po nekem standardu. V večini primerov pa imamo razhanjanja in z analizo vrzeli popišemo vsa praavila, ki jih bomo morali upoštevati, ko bomo razvijali orodje za migracijo podatkov. Preden podatke prenesemo na nov sistem, jih moramo preslikati v vrednosti oziroma obliko, ki bo na novem sistemu sprejemljiva in bo imela pomen za poslovanje.

Če to ni bilo storjeno že v koraku Analiza izvirnega sistema, se lahko določijo tudi pravila za čiščenje podatkov. Težnja naročnika je, da se prenesejo vsi podatki s starega sistema v nespremenjeni obliki, kar seveda ni mogoče. Zato je pomembno, da naročniku nazorno predstavimo posledice, ki bi jih imel prenos vseh podatkov. Najlažje je to razložiti na konkretnih podatkih, kjer ima vpogled v nejasnost določenih podatkov in vrednosti, ki so bile vpisane v star sistem brez premisleka, ali bodo vnešeni podatki imeli neko informacijo za poslovni sistem ali bodo vnesli le zmedo.

4.2.4 Analiza integracije z drugimi sistemi

Če se sistem povezuje tudi z drugimi sistemi, moramo to identificirati. Pomembno je, da ne pozabimo na podatke, ki morajo biti migrirani za pravilno delovanje integriranih sistemov tudi po preklopu na nov sistem. Ugotoviti moramo, kdo upravlja te sisteme in kdo je odgovoren za podatke v njih, saj bomo potrebovali pomoč pri določevanju pomena in preslikovanju podatkov. Tu smo imeli največ težav s komunikacijo. Lastniki podatkov integriranih sistemov so bili redko dosegljivi, včasih celo niso bili pripravljeni sodelovati. Poleg tega je bila izvedba verifikacije migriranih podatkov, do katerih so dostopali integrirani sistemi, težka in nepopolna, saj večina sistemov ni bila integrirana na testnih okoljih.

4.2.5 Določitev strategije za odklop starega sistema iz uporabe

Namen migracije je, da se z uvedbo novega sistema star sistem (večinoma) preneha uporabljati. Kot smo že ugotovili, preklon z enega sistema na drugega ni možen čez noč. Glede na zahteve naročnika in obseg migracije lahko določimo, katero strategijo bomo uporabili.[16]

Star sistem lahko:

- Odklopimo takoj, ko je migracija podatkov zaključena.
- Pustimo, da soobstaja paralelno z novim sistemom in ga izklopimo šele, ko naročnik potrdi, da nov sistem deluje po predpisanih zahtevah. Vnašanje v oba sistema:
 - lahko uporabniki izvajajo ročno,
 - lahko razvijemo rešitev, ki sinhronizira podatke iz enega v drug sistem (lahko le enosmerno, lahko pa omogočimo prenos sprememb v obeh smereh),

- lahko izvajamo migracijo podatkov po fazah, kjer z vsako fazo dodamo podatke neke nove funkcionalnosti, ločene celote ali pa izvajamo delta migracije.
- Star sistem se uporablja za vpogled v zgodovinske podatke. Sistem je lahko v uporabi, lahko pa podatke arhiviramo in jih dostavimo uporabnikom le, ko je to zahtevano. V tem primeru je za dostop do podatkov potrebno več časa.

Pri prvem naročniku smo migrirali v dveh fazah. Ker so bili podatki med različnima fazama neodvisni, smo po prvi fazi migracije onemogočili dostop do že migriranih podatkov na starem sistemu. Tako sta oba sistema obstajala, vendar sta bila neodvisna in sinhronizacija med njima ni bila potrebna. Pri drugem naročniku smo prekopili na nov sistem in ugasnili starega takoj po migraciji podatkov. Pri tretjem naročniku pa je bila migracija, tudi zaradi količine podatkov, kompleksnejša in je bilo potrebno tako razvoj aplikacije kot tudi migracijo podatkov razdeliti na več faz. V vsaki fazi smo dodajali nove integrirane sisteme, specifični podatki posameznih integriranih sistemov so bili med sabo neodvisni, star sistem pa je ostal aktiven vse do konca zadnje faze, kjer je novi sistem postal glavni sistem in je bil stari ugasnjen. Pri vsaki fazi smo zmigrirali potrebne nove podatke ali jih posodobili. Po migraciji podatkov je za sinhronizacijo na novo vnešenih ali spremenjenih podatkov med sistemoma skrbela na novo razvita funkcionalnost, ki je bila vgrajena v poslovno aplikacijo.

4.2.6 Dokončna določitev zahtev

V fazi Inicializacija smo že pridobili nekaj zahtev, vendar pa se vse do te faze seznam še dopolnjuje ali pa zahteve natančneje določijo. Pomembno je, da se v tej fazi zahteve opišejo in da se jih ne spreminja več, sicer je potrebno verifikacijo in validacijo migriranih podatkov ponoviti vsakič, ko se seznam zahtev spremeni.

V praksi se je izkazalo, da se lahko seznam zahtev spremeni tik pred migracijo v produkcijsko okolje, kljub pojasnilom naročnikom, da je spreminjanje zahtev ali dodajanje novih zahtev po fazi analize tvegano.

Pri dveh projektih je bilo potrebno ta problem reševati na nivoju vodstva, saj se naročnik ni zavedal posledic, do katerih bi nas lahko pripeljale nekontrolirane spremembe zahtev.

4.2.7 Določitev obsega migracije

Naročniku moramo nedvoumno predstaviti zahtevnost migracije glede na izbran obseg podatkov, ki naj bi se migriral, in mu pragmatično pomagati izbrati nivo podrobnosti podatkov poslovnih področij.

V primeru, da se naročnik odloči, da se migrirajo tudi zgodovinski podatki, je to potrebno upoštevati tako pri oceni tveganja projekta migracije podatkov kot tudi pri upravljanju virov.

Pri vseh treh naročnikih, kjer smo izvedli migracijo podatkov, smo nalešli na problem zaradi premajhnega zavedanja o posledicah migracije vseh podatkov. Predstava o kompleksnosti migracije se je ščasoma izoblikovala tekom projekta, saj je bila že uskladitev glede zahtev in preslikovanja osnovnih poslovnih podatkov med izvornim in ciljnim sistemom zahtevna. Nekje na sredini projekta migracije se je izkazalo, da bi za definicijo vseh potrebnih pravil preslikovanja in strategije prenosa velikih količin podatkov, če bi želeli migrirati tudi zgodovinske podatke, zmanjkalo časa.

4.2.8 Določitev strategije izvedbe migracije

Strategija izvedbe migracije je odvisna od zahtev naročnika. V primeru, da podatkov ni veliko, ni potrebno komplicirati, podatke se lahko enostavno prenese naenkrat na dan, ko se vpelje nov sistem. Ko pa imamo opravka z več podatki, je potrebno razmisliti o dobri strategiji, da ne bomo zaradi težav z zmogljivostjo ogrozili vpeljavo novega sistema.

Če je mogoče, se večina podatkov migrira še pred dnevom vpeljave novega

sistema. Tako zmanjšamo okno nedelovanja sistema na dan vpeljave novega sistema, saj se takrat zmigrirajo le novi ali spremenjeni podatki (delta migracija podatkov).

Preden se izvede migracija podatkov po izbrani strategiji, je potrebno proces sprobat tudi na testnem okolju, ki je čim bolj podoben produkcijskemu okolju, sicer nismo prepričani, da bo migracija podatkov kot proces sploh uspešna.

4.2.9 Določitev strategije testiranja migracije

Ponavadi je količina migriranih podatkov prevelika, da bi lahko stestirali vsak podatek. Zato moramo pametno določiti reprezentativno množico podatkov, ki jo bomo verificirali. To določimo glede na obseg migracije in različnosti podatkov, ki se bodo migrirali.

Testiranje migracije lahko razdelimo na tri dele:

- Testiranje pravilnosti procesa migracije podatkov
- Testiranje pravilnosti izvoženih podatkov
- Testiranje pravilnosti uvoženih podatkov

Ko je razlog za migracijo nadgradnja produkta in je izvajalcu migracije podatkov znan tako izvorni kot ciljni sistem, lahko vsa tri testiranja izvaja izvajalec.

Ko pa naročnik najame izvajalca kot tretjo osebo za izvedbo migracije in ko izvožene podatke dostavi nekdo, ki ni zadolžen tudi za izvedbo migracije, je potrebno testiranje izvoženih podatkov izvesti preden izvajalec migracije podatkov dobi izvožene podatke. V tem primeru je za testiranje podatkov zadolžena testna skupina na strani tistega, ki je izvozil podatke. Morris je v svojem delu [11] definiral območje, ki je postavljeno med tistim, ki je izvozil podatke, in tistim, ki bo podatke uporabil pri migraciji. To območje je poimenoval demilitarizirano območje (angl. *Demilitarised zone*). Opisal ga je kot mesto v procesu migracije, ko izvoženi in verificirani podatki s strani

skupine za izvoz podatkov čakajo na prevzem s strani izvajalca migracije. Z označitvijo demilitariziranega območja določimo, do katerega koraka v projektu migracije je določen udeleženec projekta odgovoren za podatke. S tem zmanjšamo nejasnosti pri odgovornosti in omejimo možne napake. Pred samo migracijo se tudi na strani izvajalca migracije preveri, da so podatki v skladu z dogovorom (tip in oblika) in da vrednosti ne odstopajo od predvidenih.

Strategija testiranja sledi strategiji izvajanja migracije. Če je načrtovano, da se bo izvedla najprej celotna migracija in nato še dve delta migraciji, moramo izvesti tudi testiranje na celotnem naboru podatkov ter nato še verificirati podatke po obeh delta migracijah.

Čeprav samo testiranje migriranih podatkov precej spominja na testiranje pri razvoju aplikacije, pa je treba upoštevati, da projekt migracije ni edina aktivnost, ki se izvaja v našem primeru. Čeprav smo rekli, da je projekt migracije samostojen projekt, pa je še kako odvisen od krovnega projekta razvoja novega sistema, kar je potrebno upoštevati tudi pri načrtovanju in zaporedju aktivnosti migracije podatkov.

Izkazalo se je, da je pri izvedbi testnih migracij potrebno posvetiti posebno pozornost okoljem, na katerih naj bi se izvajalo testno migriranje podatkov. Namreč, tudi aktivnosti krovne projekta se izvajajo na določenih okoljih in če sosledje aktivnosti krovne projekta in projekta migracije ni pravilno določeno, se lahko zgodi, da je potrebno podatke po testni migraciji podatkov pobrisati, da bi se lahko izvedlo recimo obremenitveno testiranje aplikacije, in to kljub temu, da migrirani podatki še niso bili verificirani.

Že na samem začetku migracije moramo vsaj okvirno vedeti, kdaj in na kakšen način se bo izvajalo testno migriranje podatkov ter koliko časa bo trajalo testiranje po izvedbi migracije.

Pri načrtovanju korakov migracije podatkov se je dobro zavedati, da lahko krovni projekt zamuja, kar moramo upoštevati pri pripravi plana za migracijo.

4.2.10 Odobritev s strani vodij poslovnih področij

Kot na koncu prve faze je tudi na koncu druge potrebno pridobiti soglasje odgovornih, da je bila faza uspešno zaključena in da lahko nadaljujemo z naslednjo fazo.

Seznam dokumentov, ki so značilni za to fazo prikazujeta Tabela 4.4 in Tabela 4.5.

Korak	Dokument
Analiza izvirnega sistema	Dokument z opisom izvirnega podatkovnega modela, možnih vrednosti podatkov, poslovnih objektov in poslovnih procesov
Analiza ciljnega sistema	Dokument z opisom ciljnega podatkovnega modela, možnih vrednosti podatkov, poslovnih objektov in poslovnih procesov
Analiza vrzeli	Popis vseh preslikav med izvirnim in ciljnim podatkovnim modelom, pravila za preoblikovanje in čiščenje podatkov
Analiza integracije z drugimi sistemi	Dokument s seznamom integriranih sistemov ter opisom povezav med izvirnim in integriranimi sistemi
Določitev strategije za odklop starega sistema iz uporabe	Dokument s postopkom odklopa starega sistema, ko je migracija uspešno opravljena
Dokončna določitev zahtev	Izpopolnjena in hkrati končna oblika dokumenta s seznamom zahtev (poslovnih, tehničnih in operativnih) za migracijo podatkov ter seznam omejitev

Tabela 4.4: Dokumenti faze Analiza (1. del)

Korak	Dokument
Določitev obsega migracije	Popis podatkov, ki se bodo migrirali, in njihova statistika (število vseh podatkov, število podatkov z določeno vrednostjo, itd.) ter popis podatkov, ki se ne bodo migrirali
Določitev strategije izvedbe migracije	Dokument z zbranimi koraki, kako se bo migracija izvedla (Ali bo vse v enem koraku? Ali bo inkrementalno? Če bo inkrementalno, koliko iteracij bomo imeli?)
Določitev strategije testiranja migracije	Dokument z opisom, kako bo reprezentativen vzorec podatkov za verifikacijo izbran, kakšen bo postopek migracije, kdo vse bo udeležen pri testiranju
Odobritev s strani vodij poslovnih področij	Pisna odobritev za nadaljevanje projekta migracije podatkov

Tabela 4.5: Dokumenti faze Analiza (2. del)

Vloge udeležencev na projektu pri fazi Analiza so razvidne iz Tabele 4.6 in Tabele 4.7.

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemski administrator
Analiza izvornega sistema	*	*			*		*						
Analiza ciljnega sistema			*			*		*				*	
Analiza vrzeli	*	*	*										
Analiza integracije z drugimi sistemi	*	*	*										
Določitev strategije za odklop starega sistema iz uporabe	*	*	*	*								*	*

Tabela 4.6: Koraki v fazi Analiza in vloge udeležencev na projektu (1. del)

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemiški administrator
Dokončna določitev zahtev	*	*	*	*								*	*
Določitev obsega migracije	*	*	*	*								*	*
Določitev strategije izvedbe migracije	*	*	*	*	*	*	*	*	*	*	*	*	*
Določitev strategije testiranja migracije	*	*			*	*			*	*	*	*	*
Odobritev s strani vodij poslovnih področij	*												

Tabela 4.7: Koraki v fazi Analiza in vloge udeležencev na projektu (2. del)

4.3 Faza 3: Razvoj

Seznam korakov faze Razvoj je viden na Sliki 4.4.

DNEVI			
	14.mar		21.mar
MIGRACIJA			
Razvoj	Načrtovanje	Razvoj programske opreme za migracijo podatkov (modeliranje podatkov, pravila preslikav & čiščenje podatkov)	Testiranje

Slika 4.4: Faza Razvoj.

Ta faza je enaka projektu razvoja programske opreme. Sledi lahko klasičnemu waterfall ali agilnemu pristopu, tako lahko z iterativnim izvajanjem aktivnosti gradnje orodja in testiranja pripeljemo izdelek do končne podobe.

4.3.1 Načrtovanje

V fazi načrtovanja določimo:

- Kaj mora znati početi programska oprema oziroma orodje za migracijo podatkov?
- Kje se nahajajo vhodni podatki?
- Kako preverimo, da imamo vse podatke, ki so potrebni, na voljo, tako s starega sistema kot tudi vnešene v novem sistemu?
- Katera že določena pravila preslikave podatkov med starim in novim sistemom moramo vključiti?
- Katera so pravila za potencialno čiščenje podatkov, če nismo tega storili že prej?
- Kam bomo prenesli preoblikovane podatke?
- Kako bomo preverili, da smo pravilno prenesli vse podatke?
- Kako bo izgledala programska oprema oziroma orodje za migracijo podatkov?

Lahko se odločimo tudi, da bomo uporabili že obstoječe orodje. V tem primeru moramo preveriti, ali orodje pokrije vse zahteve, in če ne, razviti le del, ki ni pokrit.

4.3.2 Razvoj programske opreme za migracijo podatkov

Glede na zahteve in omejitve pridobljene v prejšnjih fazah ter določena pravila za preslikavo in preoblikovanje podatkov se razvije orodje, ki bo uporabljeno za migracijo podatkov.

Razvoj kompleksnejšega orodja lahko razbijemo na manjše naloge.

Za naš produkt smo pripravili paket poizvedb SQL, ki smo ga prilagajali različnim strankam in fazam.

4.3.3 Testiranje

Tu je mišljeno predvsem testiranje orodja, ki ga izvaja razvijalec. Za to so potrebni neki (lahko) naključni podatki, s katerimi preverimo, da orodje deluje pravilno. Testno izvajanje migracij in verifikacija migriranih podatkov se izvaja v četrti fazi (Testiranje).

4.3.4 Odobritev s strani vodij poslovnih področij

Kot del dobre prakse je, da odgovorni glede na rezultate fazeodobrijo nadaljevanje z naslednjo fazo.

Seznam dokumentov, ki so značilni za to fazo prikazuje Tabela 4.8.

Korak	Dokument
Načrtovanje in Razvoj programske opreme za migracijo podatkov	Dokumentacija razvitega orodja
Odobritev s strani vodij poslovnih področij	Pisna odobritev za nadaljevanje projekta migracije podatkov

Tabela 4.8: Dokumenti faze Razvoj

Vloge udeležencev na projektu pri fazi Razvoj so razvidne iz Tabele 4.9.

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemski administrator
Načrtovanje			*		*							*	
Razvoj programske opreme za migracijo podatkov					*							*	
Testiranje					*							*	
Odobritev s strani vodij poslovnih področij	*												

Tabela 4.9: Koraki v fazi Razvoj in vloge udeležencev na projektu

4.4 Faza 4: Testiranje

Seznam korakov faze Testiranje je viden na Sliki 4.5.

DNEVI					
	21.mar	28.mar	4.apr	11.apr	
MIGRACIJA					
Testiranje	Izvoz izvornih testnih podatkov				
	Testno izvajanje migracije podatkov				
	Testiranje	Testiranje	Testiranje	Testiranje	

Slika 4.5: Faza Testiranje.

4.4.1 Izvoz izvornih testnih podatkov

V začetnih fazah testnega izvajanja migracije podatkov lahko uporabimo naključno generirane podatke. Kasneje pa je dobro, da so podatki izvoženi iz izvornih podatkovnih skladišč. Preden se izvede katerakoli sprememba nad podatki, morajo biti izvoženi podatki verificirani.

4.4.2 Testno izvajanje migracije podatkov

Za zagotovitev večje verjetnosti za uspešno izvedbo migracije podatkov moramo proces izvesti na podatkih in okolju, ki je čim bolj podobno produkcijskemu. Najlažje za razvoj orodja za migracijo podatkov, testiranje izvedbe migracije, verifikacijo migriranih podatkov in tako zmanjšanje tveganja za končni neuspeh je, da se aktivnosti izvajajo na podatkih iz produkcije. Žal večinoma to ni mogoče.

V zgodnjih fazah se migracija podatkov izvaja na testno generiranih podatkih oziroma še bolje, če lastnik podatkov to dopušča, na anonimiziranih podatkih, pridobljenih iz produkcijskega okolja.

Anonimizirani podatki so boljši od generiranih, saj so preoblikovani podatki s produkcijskega okolja, generirani pa so povsem naključni. Žal tudi z ano-

nimiziranimi produkcijskimi podatki ne bo mogoče preveriti, da proces migracije zajame vse vrednosti, na katere bomo naleteli med končno migracijo, še vedno pa bomo lahko preverili, da se podatki prenesejo pravilno in na pravo mesto. Zaobjeli bomo tudi možne kombinacije obstoječih poslovnih podatkov.

Problem lahko nastane, ko lastniki podatkov ne želijo omogočiti vpogleda v produkcijske podatke, niti anonimizirane, saj so podatki veliko vredni. V takem primeru se tveganje za končni neuspeh močno poveča, saj težko predvidimo, kakšne vrednosti ali obliko ima nek podatek na izvornem sistemu.

Izdelava izvedbenega projektnega plana

V okviru testiranja izvedbe migracije se popišejo vsi koraki, ki so potrebni za celovito izvedbo migracije podatkov. Zabeleži se tudi vse, kar je potrebno preveriti pred, med in po migraciji podatkov.

Seznam korakov se pripravi predvsem zato, ker v času migracije podatkov v produkcijsko okolje nimamo veliko časa za razmišljanje. Ko imamo seznam korakov pred sabo, jim lahko sledimo in tako pripeljemo migracijo do konca brez nepotrebnega stresa. Ker so v plan vključeni vsi udeleženci projekta migracije podatkov in so časovni termini korakov in odgovornosti točno določeni, se verjetnost, da bomo kakšen korak pozabili izvesti, zmanjša. V koraku planiranja se določi odgovorno osebo, ki skrbi, da so vsi udeleženci projekta obveščeni o korakih, zadolžitvah in časovnih okvirih posameznih korakov, skrbi, da se med izvajanjem migracije v produkcijsko okolje sledi korakom in sproti beleži trenutno stanje koraka (začetek koraka, uspešno izvršen korak, neuspešno izvršen korak).

Če imamo možnost opraviti testno migracijo podatkov na sistemu in podatkih, ki so čim bolj podobni sistemu in podatkom v produkcijskem okolju, bo čas izvedbe posameznih korakov zelo podoben tistim pri migraciji podatkov v produkcijsko okolje.

V kolikor naletimo na nepredviden problem, ga rešimo, zabeležimo dogodek kot namig za naslednjo migracijo in s tem gradimo bazo znanja. Po odpra-

vljeni napaki sledimo naslednjemu koraku na seznamu.

Pri naših migracijah smo seznam prilagajali naročnikom in fazam migracije, vendar se zaporedje glavnih korakov ni veliko spremenil. Tako smo z vsako novo migracijo izpopolnjevali seznam korakov, ki so potrebni za uspešno izvedbo migracije podatkov.

4.4.3 Testiranje

Pomemben in zahteven korak pri migraciji podatkov je verifikacija migriranih podatkov. Največji izziv predstavlja pametno izbrana reprezentativna podmnožica podatkov, nad katerimi se bo izvajala verifikacija. Predvsem zato, ker je ponavadi podatkov dosti več, kot je časa za izvedbo testiranja.

Testiranje celovitosti podatkov

Preveri se, ali so se vsi zahtevani podatki in podatki, ki so potrebni s poslovnega stališča, prenesli na nov sistem.

Testiranje konsistentnosti podatkov

Da bo poslovni sistem pravilno deloval, morajo imeti prenešeni podatki enak poslovni pomen kot so ga imeli na starem sistemu. Poleg tega se morajo odvisnosti med podatki obdržati.

Testiranje izgleda podatkov

Da bi naročnik lahko prevzel in uporabniki sprejeli nov sistem, mora biti tudi izgled podatkov v aplikaciji tak, kot je bilo določeno z zahtevami. Vsako odstopanje je potrebno preveriti s poslovnimi analitiki.

Testiranje procesiranja podatkov

Da lahko nov sistem pravilno deluje, morajo biti vsi prenešeni podatki v skladu z zahtevami, sicer bi lahko naleteli celo na izpad novega sistema,

kar bi lahko ogrozilo uspešnost migracije podatkov in s tem tudi uspešnost uvedbe novega sistema.

Integracijski test

Če je nov sistem povezan tudi z drugimi sistemi, je potrebno preveriti, da so prenešeni podatki ustrezni tudi za druge sisteme.

4.4.4 Odobritev s strani vodij poslovnih področij

Ko so odgovorni zadovoljni z izzidom te faze, odobrijo nadaljevanje projekta.

Seznam dokumentov, ki so značilni za to fazo prikazuje Tabela 4.10.

Korak	Dokument
Izvoz izvornih testnih podatkov	Poročilo z rezultati verifikacije izvoženih podatkov
Testno izvajanje migracije podatkov	Rezultati obremenitvenega testiranja, seznam morebitnih problematičnih podatkov ter izvedbeni projektni plan
Testiranje	Poročilo z rezultati testiranja celovitosti, konsistentnosti, izgleda, procesiranja podatkov ter rezultat integracijskih testov
Odobritev s strani vodij poslovnih področij	Pisna odobritev za nadaljevanje projekta migracije podatkov

Tabela 4.10: Dokumenti faze Testiranje

Vloge udeležencev na projektu pri fazi Testiranje so razvidne iz Tabele 4.11 in Tabele 4.12.

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemi administrator
Izvoz izvornih testnih podatkov			*		*		*		*			*	
Testno izvajanje migracije podatkov												*	
Izdelava izvedbenega projektnega plana	*	*	*	*		*		*	*	*	*	*	*
Testiranje celovitosti podatkov										*			
Testiranje konsistentnosti podatkov										*			

Tabela 4.11: Koraki v fazi Testiranje in vloge udeležencev na projektu (1.del)

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemske administrator
Testiranje izgleda podatkov				*						*			
Testiranje procesiranja podatkov										*			
Integracijski test										*			
Odobritev s strani vodij poslovnih področij	*												

Tabela 4.12: Koraki v fazi Testiranje in vloge udeležencev na projektu (2.del)

4.5 Faza 5: Izvedba

Seznam korakov faze Izvedba je viden na Sliki 4.6.

DNEVI									
	11.apr	18.apr	25.apr		2.maj		9.maj	16.maj	23.maj
MIGRACIJA									
Izvedba	Zadnja testna izvedba migracije podatkov	Testiranje zadnje testne izvedbe migracije podatkov	Pridobitev soglasja vodij poslovnih področij za migracijo v produkcijsko okolje	Izvoz izvirnih podatkov	Izvedba migracije podatkov	Validacija migriranih podatkov	Vzpostavitev prejšnjega stanja izvirnega sistema	Predaja novega sistema naročniku in podatkov njihovim lastnikom	Nadzorovanje po izvedbi migracije
								Preklop na nov sistem	

Slika 4.6: Faza Izvedba.

4.5.1 Zadnja testna izvedba migracije podatkov

Kot je bilo že velikokrat poudarjeno, je dobro, da se pred dejansko izvedbo migracije podatkov v produkcijsko okolje izvede migracija na podobnem okolju na realnih podatkih ali vsaj anonimiziranih realnih podatkih. S tem potrdimo, da bo izvedba migracije uspešna tudi v produkcijskem okolju. Žal to ni vedno mogoče, zato zna biti izvedba migracije podatkov v produkcijsko okolje stresna in negotova. V takem primeru je potrebno naročniku in vsem vpletenim na projektu razvoja novega sistema jasno razložiti, da je izzid migracije brez testiranja na pravih podatkih vprašljiv in da morajo prevzeti odgovornost za morebitne probleme, ki se lahko pojavijo zaradi nepredvidenih podatkov. V tem koraku se priporoča sledenje izvedbenemu projektnemu planu, da se tako verificira seznam korakov in njihov pravilni vrstni red, ki ga bomo izvedli tudi pri migraciji podatkov v produkcijsko okolje. V primeru odstopanja se izvedbeni projektni plan popravi.

4.5.2 Testiranje zadnje testne izvedbe migracije podatkov

S tem korakom pripravimo poročilo za vodje poslovnih področij, da se ti lahko odločijo, ali bodo odobrili končen korak, dejansko migracijo podatkov

v produkcijsko okolje.

Če soglasja za migracijo podatkov v produkcijsko okolje ne dobimo, moramo skupaj z vodjami poslovnih področij določiti, kje so problemi in kako jih lahko rešimo. Če je potrebno, se popravi orodje za migracijo podatkov (pravila za preslikavo in preoblikovanje podatkov) in izvede ponovno čiščenje podatkov. Nato se ponovno izvede zadnja testna migracija podatkov v dogovorjenem terminu in se počaka na soglasje vodij poslovnih področij za nadaljevanje aktivnosti na projektu.

4.5.3 Pridobitev soglasja vodij poslovnih področij za migracijo v produkcijsko okolje

Na podlagi prejšnjega koraka vodje poslovnih področij odobrijo nadaljevanje z naslednjimi koraki. V primeru, da nadaljevanje ni odobreno, je potrebno napake popraviti in ponovno opraviti testno migracijo in testiranje.

4.5.4 Izvoz izvornih podatkov

Po dogovorjeni varnostni proceduri skrbnik podatkovnih baz (DBA) izvozi podatke iz izvornih podatkovnih skladišč, tester pa verificira njihovo pravilnost. Za tem lahko sledi izvedba migracije podatkov.

4.5.5 Izvedba migracije podatkov

Če je sistem ugasnjen med izvedbo migracije podatkov (in uvedbo novega sistema), moramo o tem obvestiti uporabnike sistema. Tudi ko sistem deluje med migracijo, moramo obvestiti uporabnike, da so ti pozorni, če pride do kakšne napake, in to takoj javijo, da se problem razišče in odpravi.

Preden karkoli spremenimo na sistemu, moramo poskrbeti za izdelavo varnostnih kopij podatkov. Migracija se izvede po izbrani strategiji. Med izvedbo migracije se sledi izvedbenemu projektному planu.

Po izvedbi migracije podatkov je sistem lahko še nek čas neaktiven, vendar dostopen testni ekipi, ki lahko verificira podatke, in vodjam poslovnih

področij za validacijo podatkov.

Pri vsaki izvedbi v produkcijsko okolje pri nas je odgovorni za to nalogo vodil proces od začetka do konca ter beležil stanje izvedbe posameznih korakov. Ko je bil nek korak končan, se je to označilo na seznamu korakov, nato pa se je proces nadaljeval. Tako smo imeli zabeleženo, kaj se je izvedlo in koliko časa je trajalo. Poleg tega smo dokumentirali tudi probleme pri izvedbi.

Migracija podatkov je kompleksen proces in težko je predvideti vse probleme, ki bi jih lahko srečali. Zato se velikokrat zgodi, da moramo po končani migraciji, ko je sistem pripravljen za uporabo, popraviti nekatere podatke. Zahtevnost popravkov je nepredvidljiva, zato je pomembno, da se pred izvedbo migracije v produkcijo preveri čim več možnih preslikav na čim bolj različnih podatkih.

Pri vsakem projektu migracije podatkov smo naleteli na probleme po vplelji novega sistema. V nekaterih primerih ni bilo pravilno določeno iz katerega podatkovnega skladišča naj se izvozijo podatki. Zgodilo se je tudi, da matematična formula (poslovno pravilo) za izračun zneska ni bila pravilno določena. Prav tako se je zgodilo, da poslovno pravilo ni bilo upoštevano in se je tako preneslo manj podatkov kot bi se moralo. Najtežji primer pa je bil, ko podatki niso bili dostavljeni kot dogovorjeno, temveč je bil format drugačen. Tudi po treh iteracijah dostave izvoženih podatkov format še vedno ni bil pravilen. Na koncu smo prekoračili naše zadolžitve ter popravili format sami, kar je bilo sicer hitreje, ne pa tudi v skladu z dogovorom o odgovornosti. Seveda so se problemi pojavili tudi na strani vnosa izvoženih podatkov. Ker se je naš produkt neprestano nadgrajeval (dodajale so se nove funkcionalnosti) in so se seznam podatkov potrebnih za pravilno delovanje sistema in privzete vrednosti spremenili od prve opravljene migracije podatkov do zadnje, smo vnesli podatke, ki niso bili potrebni ali pa smo manjkajoče vrednosti nadomestili z napačnimi privzetimi vrednostmi.

4.5.6 Validacija migriranih podatkov

Vodje poslovnih področij morajo po koncu migracije validirati rezultat migracije. Če odločijo, da je bila migracija podatkov uspešno izvršena, sledi korak predaje sistema naročniku in nato preklon na nov sistem, sicer pa je potrebno vzpostaviti prejšnje stanje sistema in poskusiti uspešno opraviti migracijo kasneje, ko so napake odpravljene in ko je rezultat migracije pravilen.

4.5.7 Vzpostavitev prejšnjega stanja izvirnega sistema

Če se migracija podatkov ne zaključi uspešno in podatkov ni moč popraviti v določenem času, je potrebno sistem vrniti na prejšnje stanje oziroma priklopiti star sistem nazaj. Proces zna biti težaven, zato je pomembno, da korake popišemo tudi v izvedbenem projektnem planu, saj takrat nimamo časa za podroben razmislek.

4.5.8 Predaja novega sistema naročniku in podatkov njihovim lastnikom

Po uspešni migraciji lahko nov sistem predamo lastnikom. Varnostne pravice, ki smo jih imeli med časom izvedbe migracije podatkov, se lahko ponastavijo glede na varnostno politiko naročnika. Lastniki podatkov prevzamejo lastništvo nad migriranimi podatki.

4.5.9 Preklon na nov sistem

Preklon na nov sistem se izvede skladno s strategijo preklopa na nov sistem, ki smo ga določili v drugi fazi.

4.5.10 Nadzorovanje po izvedbi migracije

Odvisno od podpisane pogodbe in dogovorov, se lahko zgodi, da tudi po uspešno opravljeni migraciji delo še ni končano. Po uvedbi novega sistema v uporabo je potrebno še vedno nadzorovati delovanje sistema. Nekatere

napake pri določitvi preslikave podatkov med starim in novim sistemom ali pri samem procesu migracije podatkov se lahko pokažejo šele kasneje, celo več mesecev po uvedbi sistema.

Seznam dokumentov, ki so značilni za to fazo prikazuje Tabela 4.13.

Korak	Dokument
Zadnja testna izvedba migracije podatkov	Poročilo o izvedbi zadnje testne migracije
Testiranje zadnje testne izvedbe migracije podatkov	Poročilo z rezultati testiranja celovitosti, konsistentnosti, izgleda, procesiranja podatkov ter rezultat integracijskih testov
Izvoz izvornih podatkov	Poročilo z rezultati verifikacije izvoženih podatkov
Izvedba migracije podatkov	Poročilo o izvedbi migracije
Validacija migriranih podatkov	Poročilo o uspešni ali neuspešni validaciji migriranih podatkov
Vzpostavitev prejšnjega stanja izvornega sistema	V primeru neuspešne validacije, poročilo o vzpostavitvi prejšnjega stanja
Predaja novega sistema naročniku in podatkov njihovim lastnikom	V primeru uspešne validacije, poročilo o predaji novega sistema
Preklop na nov sistem	V primeru uspešne validacije, poročilo o preklopu na nov sistem in njegovem delovanju
Nadzorovanje po izvedbi migracije	Poročilo z različnimi statističnimi podatki o uporabi migriranih podatkov

Tabela 4.13: Dokumenti faze Izvedba

Vloge udeležencev na projektu pri fazi Izvedba so razvidne iz Tabele 4.14 in Tabele 4.15.

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemi administrator
Zadnja testna izvedba migracije podatkov	*												
Testiranje zadnje testne izvedbe migracije podatkov	*	*	*	*	*	*	*	*				*	
Izvoz izvornih podatkov	*	*	*	*								*	*
Izvedba migracije podatkov	*	*	*	*								*	*
Validacija migriranih podatkov	*	*	*	*								*	*

Tabela 4.14: Koraki v fazi Izvedba in vloge udeležencev na projektu (1.del)

Korak	Vodja projekta	Vodje poslovnih področij	Poslovni analitik na strani vira podatkov	Poslovni analitik na strani izvajalca migracije	Razvijalec podatkovnih baz, na strani vira podatkov	Razvijalec podatkovnih baz	Razvijalec starega poslovnega sistema	Razvijalec novega poslovnega sistema	Tester na strani izvora podatkov	Tester na strani izvajalca migracije	Skupina za upravljanje s konfiguracijami	Skrbnik podatkovnih baz	Sistemiški administrator
Vzpostavitev prejšnjega stanja izvornega sistema			*		*		*		*		*	*	*
Predaja novega sistema naročniku in podatkov njihovim lastnikom	*	*										*	
Preklop na nov sistem						*		*			*	*	*
Nadzorovanje po izvedbi migracije				*		*	*			*		*	*

Tabela 4.15: Koraki v fazi Izvedba in vloge udeležencev na projektu (2.del)

Poglavje 5

Sklepne ugotovitve

Praktična migracija podatkov [11], Procesni model [2] in Metoda migracije podatkov za podatkovno intenzivno produktno programsko opremo [7] zelo dobro pokrivajo svoj del področja. Morrisove izkušnje pri izvajanju migracije in podrobna razčlenitev vodenja projekta migracije podatkov olajša višjenivojsko obravnavo migracije podatkov. Idu ter Matthes, Schultz in Haller pa so strnili procesne korake migracije podatkov v uporaben paket, ki ga je možno v veliki meri uporabiti na večini projektov.

Z analizo literature se je tudi pri naši praktični izvedbi migracije podatkov proces bistveno izboljšal, predvsem glede dokumentiranja posameznih korakov, pa tudi bolj tekočega izvajanja zaporedja korakov. Različna opozorila in nasveti avtorjev glede problemov so izboljšala tudi obvladovanje tveganj, komunikacijo na projektu in razdelitev odgovornosti med udeležence projekta. Zanimivo je, da tudi danes nova literatura o migraciji podatkov v primerjavi z drugimi področji v računalništvu zelo počasi nastaja. Verjetno je to povezano z dejstvom, da se migracija podatkov ne izvaja vsak dan in da je ponavadi projekt migracije podatkov dolgotrajen, kar pomeni, da je potreben čas za nova dognanja in izkušnje ter rezultate.

Literatura

- [1] F. Fleurey et al. “*Model-Driven Engineering for Software Migration in a Large Industrial Context*”. V *ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems*, Nashville, TN, ZDA, 2007 [Online]. Dosegljivo: <http://www.irisa.fr/triskell/publis/2007/fleurey07a.pdf>. [Dostopano: 2. 8. 2016].
- [2] K. Haller, F. Matthes in C. Schulz. “*A detailed process model for large scale data migration projects*”. V *15th International Conference on Business Information Systems*, BIS 2012, Vilnius, Litva, 21.-23. maj, 2012 [Online]. Dosegljivo: http://www.klaushaller.net/media/BIS2012_halller_matthes_schulz_data_migration_process_model.pdf. [Dostopano: 2. 8. 2016].
- [3] K. Haller, F. Matthes in C. Schulz. “*Testing & Quality Assurance in Data Migration Projects*”. V *27th IEEE International Conference on Software Maintenance*, Williamsburg, VA, ZDA, 25.-30. september, 2011 [Online]. Dosegljivo: http://www.klaushaller.net/media/icsm11_testing_data_migration.pdf. [Dostopano: 2. 8. 2016].
- [4] K. Haller. “*Towards the Industrialization of Data Migration: Concepts and Patterns for Standard Software Implementation Projects*”. V *CAISE 2009*, Amsterdam, Nizozemska, 2009 [Online]. Dosegljivo:

- http://www.klaushaller.net/media/haller_datamigration_caise2009.pdf. [Dostopano: 2. 8. 2016].
- [5] J. Henrard et al. “*Strategies for Data Reengineering*”. V *FNRS contact day on «Software (re-)engineering»*, Louvain-la-Neuve, Belgija, 22. maj, 2003 [Online]. Dosegljivo: <https://pure.fundp.ac.be/ws/files/196526/FNRS-ReEngineering.pdf>. [Dostopano: 2. 8. 2016].
- [6] J. Henrard et al. “*Strategies for Data Reengineering*”. V *9th Working Conference on Reverse Engineering (WCRE 2002)*, Richmond, VA, ZDA, 28. oktober-1. november, 2002 [Online]. Dosegljivo: <https://staff.info.unamur.be/dbm/Documents/Publications-LIBD/Articles-Conferences/conferences-WCRE/WCRE-2002/WCRE-2002.pdf>. [Dostopano: 2. 8. 2016].
- [7] A. Idu. “*Data Migration for Data Intensive Software Products*”. Magistrsko delo, Utrecht University, Nizozemska, 2012 [Online]. Dosegljivo: http://www.staff.science.uu.nl/~khadk101/publication/Idu_Thesis.pdf. [Dostopano: 2. 8. 2016].
- [8] V. Khusidman in W. Ulrich. “*Architecture-Driven Modernization: Transforming the Enterprise*”. 2007 [Online]. Dosegljivo: http://adm.omg.org/ADMRoadmapv6_VK.pdf. [Dostopano: 2. 8. 2016].
- [9] R. Levine. “*Data migration strategies*”. (17. avgust, 2009). wikibon [Online]. Dosegljivo: <http://wikibon.org/wiki/v/Data%20migration%20strategies>. [Dostopano: 2. 8. 2016].
- [10] D. Mistry. “*Big Bang vs Phased – the ideal migration approach*”. (21. november, 2014). xceedgroup [Online]. Dosegljivo: <http://www.xceedgroup.com/xceed-blog/big-bang-vs-phased-the-ideal-migration-approach>. [Dostopano: 2. 8. 2016].

-
- [11] J. Morris. *Practical Data Migration*, druga izdaja. Swindon, Velika Britanija: BCS The Chartered Institute for IT, 2012.
- [12] S. Tammer. “*Renaissance: A method to support Software System Evolution*”. (13. april 2012). Utrecht University [Online]. Dosegljivo: <http://foswiki.cs.uu.nl/foswiki/MethodEngineering/RenaissanceAMethodtoSupportSoftwareSystemEvolution20112012>. [Dostopano: 2. 8. 2016].
- [13] B. Wu et al. “*The Butterfly Methodology : A Gateway-free Approach for Migrating Legacy Information Systems*”. V *3rd IEEE Conference on Engineering of Complex Computer Systems (ICECCS97)*, Villa Olmo, Como, Italija. 8.-12. september, 1997. str. 200-205, IEEE Computer Society, 1997 [Online]. Dosegljivo: <http://www.dtic.upf.edu/~jbisbal/publications/iceccs97.pdf>. [Dostopano: 2. 8. 2016].
- [14] B. Wu et al. “*Legacy Systems Migration - A Method and its Tool-kit Framework*”. V *Joint 1997 Asia Pacific Software Engineering Conference and International Computer Science Conference*, Hong Kong, Kitajska. 2.-5. december, 1997. str. 312-320, Ed. IEEE Computer Society, 1997 [Online]. Dosegljivo: <http://www.dtic.upf.edu/~jbisbal/publications/icsc97.pdf>. [Dostopano: 2. 8. 2016].
- [15] ETL Solutions. “*Preparing a Data Migration Plan: A practical introduction to data migration strategy and planning.*”. (november 2012) [Online]. Dosegljivo: <http://www.etlsolutions.com/wp-content/uploads/2012/11/Data-Migration-eBook.pdf>. [Dostopano: 2. 8. 2016].
- [16] Data Migration Pro Expert Journal. “*The data migration go-live strategy - what is it and why does it matter?*”. (26. marec 2009) [Online]. Dosegljivo:

<http://www.datamigrationpro.com/data-migration-articles/2009/3/26/the-data-migration-go-live-strategy-what-is-it-and-why-does.html>. [Dostopano: 2. 8. 2016].

- [17] Phoecho. “*Data Migration Methodology*”. (november 2010) [Online].

Dosegljivo:

<http://phoecho.blog.com/files/2010/11/>

Data-migration-methodology.pptx. [Dostopano: 19. 12. 2015].

- [18] Wikipedia. “*Method engineering*”. (marec 2016) [Online]. Dosegljivo:

https://en.wikipedia.org/wiki/Method_engineering. [Dostopano 2. 8. 2016].